CONTENTS

CHAPTER 1
Cisco IOS PKI Overview Understanding and Planning a PKI 1
Finding Feature Information 1
Information About Cisco IOS PKI 1
What Is Cisco IOS PKI 1
RSA Keys Overview 2
What Are CAs 3
Hierarchical PKI Multiple CAs 3
When to Use Multiple CAs 4
Certificate Enrollment How It Works 4
Certificate Enrollment Via Secure Device Provisioning 5
Certificate Revocation Why It Occurs 5
Planning for a PKI 5
Where to Go Next 6
Additional References 6
Glossary 8

CHAPTER 2
Deploying RSA Keys Within a PKI 9
Finding Feature Information 9
Prerequisites for Configuring RSA Keys for a PKI 10
Information About RSA Keys Configuration 10
RSA Keys Overview 10
Usage RSA Keys Versus General-Purpose RSA Keys 10
How RSA Key Pairs are Associated with a Trustpoint 11
Reasons to Store Multiple RSA Keys on a Router 11
Benefits of Exportable RSA Keys 11
Passphrase Protection While Importing and Exporting RSA Keys 12
How to Set Up and Deploy RSA Keys Within a PKI 12
Generating an RSA Key Pair 12
What to Do Next 13
Managing RSA Key Pairs and Trustpoint Certificates 14
Exporting and Importing RSA Keys 17
    Exporting and Importing RSA Keys in PKCS12 Files 17
    Exporting and Importing RSA Keys in PEM-Formatted Files 19
Encrypting and Locking Private Keys on a Router 22
Removing RSA Key Pair Settings 24
Configuration Examples for RSA Key Pair Deployment 26
    Generating and Specifying RSA Keys Example 26
    Exporting and Importing RSA Keys Examples 26
        Exporting and Importing RSA Keys in PKCS12 Files Example 26
        Exporting and Importing and RSA Keys in PEM Files Example 27
        Exporting Router RSA Key Pairs and Certificates from PEM Files Example 28
        Importing Router RSA Key Pairs and Certificate from PEM Files Example 29
Encrypting and Locking Private Keys on a Router Examples 29
    Configuring and Verifying an Encrypted Key Example 29
    Configuring and Verifying a Locked Key Example 30
Where to Go Next 30
Additional References 30
Feature Information for RSA Keys Within a PKI 31

CHAPTER 3
Configuring Authorization and Revocation of Certificates in a PKI 37
    Finding Feature Information 37
    Prerequisites for Authorization and Revocation of Certificates 38
    Restrictions for Authorization and Revocation of Certificates 38
    Information About Authorization and Revocation of Certificates 38
        PKI Authorization 38
        PKI and AAA Server Integration for Certificate Status 39
            RADIUS or TACACS+ Choosing a AAA Server Protocol 39
            Attribute-Value Pairs for PKI and AAA Server Integration 40
        CRLs or OCSP Server Choosing a Certificate Revocation Mechanism 41
            What Is a CRL 41
                Querying All CDPs During Revocation Check 41
            What Is OCSP 42
                When to Use an OCSP Server 42
CHAPTER 4
Configuring Certificate Enrollment for a PKI 93
Finding Feature Information 93
Prerequisites for PKI Certificate Enrollment 94
Information About Certificate Enrollment for a PKI 94
  What Are CAs 94
  Framework for Multiple CAs 94
  Authentication of the CA 95
  Supported Certificate Enrollment Methods 95
  Cisco IOS Suite-B Support for Certificate Enrollment for a PKI 96
  Registration Authorities 96
  Automatic Certificate Enrollment 97
  Certificate Enrollment Profiles 98
How to Configure Certificate Enrollment for a PKI 98
  Configuring Certificate Enrollment or Autoenrollment 98
  Configuring Manual Certificate Enrollment 104
    PEM-Formatted Files for Certificate Enrollment Request 104
    Restrictions for Manual Certificate Enrollment 104
    Configuring Cut-and-Paste Certificate Enrollment 104
    Configuring TFTP Certificate Enrollment 107
  Certifying a URL Link for Secure Communication with a Trend Micro Server 110
  Configuring a Persistent Self-Signed Certificate for Enrollment via SSL 115
    Persistent Self-Signed Certificates Overview 115
    Restrictions 116
    Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters 116
    Enabling the HTTPS Server 118
CHAPTER 5

Setting Up Secure Device Provisioning for Enrollment in a PKI 141

Finding Feature Information 141
Prerequisites for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI 142
Information About Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI 143

SDP Overview 143
How SDP Works 144

SDP Prep-Connect Phase 144
SDP Connect Phase 146
SDP Start Phase 148
SDP Welcome Phase 149
SDP Introduction Phase 149
SDP Completion Phase 150

SDP Leveraging USB Tokens 151
Use of SDP to Configure the USB Token 152
SDP Phases with a USB Token 153
Use of the Configured USB Token 154
How SDP Uses an External AAA Database 154
Authentication and Authorization Lists for SDP 154
Authentication and Authorization Lists for an Administrative Introducer 155
How Custom Templates Work with SDP 156
Contents

Custom Template Variable Expansion 156

Custom Template Variable Expansion Rules 157
  Custom HTML Template Expansion Rules 157
  URL Template Expansion Rules 157
  URL Template Expansion Rules for iPhone Deployment 157
  Custom Configuration and File Template Variable Expansion Rules 158

Default Templates for SDP Transaction Web Pages 160
  Default Prep-Connect Template 160
  Default Start Page Template 162
  Default Welcome Page Template 162
  Default Introduction Page Template 162
  Default Admin-Introduction Page Template 162
  Default Completion Page Template 163

Default Template for the Configuration File 163

How SDP Deploys Apple iPhones in a PKI 163
  SDP Registrar Deployment Phases of the Apple iPhone in a PKI 164
    Start SDP Deployment Phase 164
    Welcome SDP Deployment Phase 165
    Introduction SDP Deployment Phase 165
    Post-Introduction SDP Deployment Phase 168
    Second-Introduction SDP Deployment Phase 168
    Second Post-Introduction SDP Deployment Phase 169
    Completion SDP Deployment Phase 169

How to Set Up Secure Device Provisioning (SDP) for Enrollment in a PKI 169
  Enabling the SDP Petitioner 169
    Troubleshooting Tips 171
    What to Do Next 172
  Enabling the SDP Registrar and Adding AAA Lists to the Server 172
    Prerequisites 172
    Restrictions 172
    The template config Command 173
  Enabling the SDP Registrar for Certificate-Based Authorization 175
  Configuring the SDP Registrar to Deploy Apple iPhones 177
    Apple CA Server Trustpoint Certificate Configuration 180
  Configuring an Administrative Introducer 182
## Configuring Custom Templates 185

Configuration Examples for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI 187

Verifying the SDP Registrar Example 187

Verifying the SDP Petitioner Example 189

Adding AAA Lists to a RADIUS or TACACS+ Server Examples 192
  - TACACS+ AAA Server Database Example 192
  - RADIUS AAA Server Database Example 192
  - AAA List on a TACACS+ and a RADIUS AAA Server Example 192

Using a Configuration Template File Example 193

CGI Script Example 193

Configuring the Petitioner and Registrar for Certificate-Based Authentication Example 195

Configuring an Administrative Introducer Using Authentication and Authorization Lists Example 195

### Additional References 196

Feature Information for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI 197

## Chapter 6 Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment 203

Finding Feature Information 204

Prerequisites for Configuring a Cisco IOS Certificate Server 204

Restrictions for Configuring a Cisco IOS Certificate Server 205

Information About Cisco IOS Certificate Servers 205
  - RSA Key Pair and Certificate of the Certificate Server 205
    - How the CA Certificate and CA Key Are Automatically Archived 205

Certificate Server Database 206
  - Certificate Server Database File Storage 206
  - Certificate Server Database File Publication 207

Trustpoint of the Certificate Server 208

Certificate Revocation Lists (CRLs) 209

Certificate Server Error Conditions 209

Certificate Enrollment Using a Certificate Server 210
  - SCEP Enrollment 211

Types of CA Servers Subordinate and Registration Authorities (RAs) 211

Automatic CA Certificate and Key Rollover 211
### CHAPTER 7: Storing PKI Credentials

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finding Feature Information</td>
<td>259</td>
</tr>
<tr>
<td>Prerequisites for Storing PKI Credentials</td>
<td>260</td>
</tr>
<tr>
<td>Restrictions for Storing PKI Credentials</td>
<td>260</td>
</tr>
<tr>
<td>Information About Storing PKI Credentials</td>
<td>261</td>
</tr>
<tr>
<td>Storing Certificates to a Local Storage Location</td>
<td>261</td>
</tr>
<tr>
<td>PKI Credentials and USB Tokens</td>
<td>261</td>
</tr>
<tr>
<td>How a USB Token Works</td>
<td>261</td>
</tr>
<tr>
<td>Benefits of USB Tokens</td>
<td>262</td>
</tr>
<tr>
<td>How to Configure PKI Storage</td>
<td>263</td>
</tr>
<tr>
<td>Specifying a Local Storage Location for Certificates</td>
<td>263</td>
</tr>
<tr>
<td>Setting Up and Using USB Tokens on Cisco Devices</td>
<td>265</td>
</tr>
<tr>
<td>Storing the Configuration on a USB Token</td>
<td>265</td>
</tr>
<tr>
<td>Logging Into and Setting Up the USB Token</td>
<td>265</td>
</tr>
<tr>
<td>How RSA Keys are Used with a USB Token</td>
<td>265</td>
</tr>
<tr>
<td>Configuring the Device for Manual Login</td>
<td>265</td>
</tr>
<tr>
<td>What to Do Next</td>
<td>266</td>
</tr>
<tr>
<td>Configuring the USB Token</td>
<td>267</td>
</tr>
<tr>
<td>PINs and Passphrases</td>
<td>267</td>
</tr>
<tr>
<td>Unlocking and Locking the USB Token</td>
<td>267</td>
</tr>
<tr>
<td>Secondary Configuration and Unconfiguration Files</td>
<td>267</td>
</tr>
<tr>
<td>What to Do Next</td>
<td>270</td>
</tr>
<tr>
<td>Setting Administrative Functions on the USB Token</td>
<td>270</td>
</tr>
<tr>
<td>Troubleshooting USB Tokens</td>
<td>273</td>
</tr>
<tr>
<td>Troubleshooting the USB Port Connection</td>
<td>273</td>
</tr>
<tr>
<td>Determining if a USB Token is Supported by Cisco</td>
<td>274</td>
</tr>
<tr>
<td>Determining USB Token Device Problems</td>
<td>274</td>
</tr>
<tr>
<td>Displaying USB Token Information</td>
<td>276</td>
</tr>
</tbody>
</table>
Configuration Examples for PKI Storage 277
  Example: Storing Certificates to a Specific Local Storage Location 277
  Example: Logging Into a USB Token and Saving RSA Keys to the USB Token 277
Additional References 278
Feature Information for Storing PKI Credentials 279

CHAPTER 8
Source Interface Selection for Outgoing Traffic with Certificate Authority 285
Finding Feature Information 285
Information About Source Interface Selection for Outgoing Traffic with Certificate Authority 286
  Certificates That Identify an Entity 286
  Source Interface for Outgoing TCP Connections Associated with a Trustpoint 286
How to Configure Source Interface Selection for Outgoing Traffic with Certificate Authority 286
  Configuring the Interface for All Outgoing TCP Connections Associated with a Trustpoint 286
Troubleshooting Tips 289
Configuration Examples for Source Interface Selection for Outgoing Traffic with Certificate Authority 289
  Source Interface Selection for Outgoing Traffic with Certificate Authority Example 289
Additional References 290
Feature Information for Source Interface Selection for Outgoing Traffic with Certificate Authority 291
Glossary 292
Cisco IOS public key infrastructure (PKI) provides certificate management to support security protocols such as IP Security (IPSec), secure shell (SSH), and secure socket layer (SSL). This module identifies and describes concepts that are needed to understand, plan for, and implement a PKI.

- Finding Feature Information, page 1
- Information About Cisco IOS PKI, page 1
- Planning for a PKI, page 5
- Where to Go Next, page 6
- Additional References, page 6
- Glossary, page 8

**Finding Feature Information**

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Information About Cisco IOS PKI**

**What Is Cisco IOS PKI**

A PKI is composed of the following entities:

- Peers communicating on a secure network
• At least one certification authority (CA) that grants and maintains certificates

• Digital certificates, which contain information such as the certificate validity period, peer identity information, encryptions keys that are used for secure communications, and the signature of the issuing CA

• An optional registration authority (RA) to offload the CA by processing enrollment requests

• A distribution mechanism (such as Lightweight Directory Access Protocol [LDAP] or HTTP) for certificate revocation lists (CRLs)

PKI provides customers with a scalable, secure mechanism for distributing, managing, and revoking encryption and identity information in a secured data network. Every entity (a person or a device) participating in the secured communications is enrolled in the PKI in a process where the entity generates an Rivest, Shamir, and Adelman (RSA) key pair (one private key and one public key) and has their identity validated by a trusted entity (also known as a CA or trustpoint).

After each entity enrolls in a PKI, every peer (also known as an end host) in a PKI is granted a digital certificate that has been issued by a CA. When peers must negotiate a secured communication session, they exchange digital certificates. Based on the information in the certificate, a peer can validate the identity of another peer and establish an encrypted session with the public keys contained in the certificate.

Although you can plan for and set up your PKI in a number of different ways, the figure below shows the major components that make up a PKI and suggests an order in which each decision within a PKI can be made. The figure is a suggested approach; you can choose to set up your PKI from a different perspective.

**Figure 1: Deciding How to Set Up Your PKI**

**RSA Keys Overview**

An RSA key pair consists of a public key and a private key. When setting up your PKI, you must include the public key in the certificate enrollment request. After the certificate has been granted, the public key will be included in the certificate so that peers can use it to encrypt data that is sent to the router. The private key is
kept on the router and used both to decrypt the data sent by peers and to digitally sign transactions when negotiating with peers.

RSA key pairs contain a key modulus value. The modulus determines the size of the RSA key. The larger the modulus, the more secure the RSA key. However, keys with large modulus values take longer to generate, and encryption and decryption operations take longer with larger keys.

**What Are CAs**

A CA, also known as a trustpoint, manages certificate requests and issues certificates to participating network devices. These services (managing certificate requests and issuing certificates) provide centralized key management for the participating devices and are explicitly trusted by the receiver to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

You can use a CA provided by a third-party CA vendor, or you can use an “internal” CA, which is the Cisco IOS Certificate Server.

**Hierarchical PKI Multiple CAs**

PKI can be set up in a hierarchical framework to support multiple CAs. At the top of the hierarchy is a root CA, which holds a self-signed certificate. The trust within the entire hierarchy is derived from the RSA key pair of the root CA. The subordinate CAs within the hierarchy can be enrolled with either the root CA or with another subordinate CA. These enrollment options are how multiple tiers of CAs are configured. Within a hierarchical PKI, all enrolled peers, can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA.

The figure below shows the enrollment relationships among CAs within a three-tiered hierarchy.

*Figure 2: Three-Tiered CA Hierarchy Sample Topology*

Each CA corresponds to a trustpoint. For example, CA11 and CA12 are subordinate CAs, holding CA certificates that have been issued by CA1; CA111, CA112, and CA113 are also subordinate CAs, but their CA certificates have been issued by CA11.
When to Use Multiple CAs

Multiple CAs provide users with added flexibility and reliability. For example, subordinate CAs can be placed in branch offices while the root CA is at the office headquarters. Also, different granting policies can be implemented per CA, so you can set up one CA to automatically grant certificate requests while another CA within the hierarchy requires each certificate request to be manually granted.

Scenarios in which at least a two-tier CA is recommended are as follows:

- Large and very active networks in which a large number of certificates are revoked and reissued. A multiple tier CA helps to control the size of the CRLs.
- When online enrollment protocols are used, the root CA can be kept offline with the exception of issuing subordinate CA certificates. This scenario provides added security for the root CA.

Certificate Enrollment How It Works

Certificate enrollment is the process of obtaining a certificate from a CA. Each end host that wants to participate in the PKI must obtain a certificate. Certificate enrollment occurs between the end host requesting the certificate and the CA. The figure below and the following steps describe the certificate enrollment process.

**Figure 3: Certificate Enrollment Process**

1. The end host generates an RSA key pair.
2. The end host generates a certificate request and forwards it to the CA (or the RA, if applicable).
3. The CA receives the certificate enrollment request, and, depending on your network configuration, one of the following options occurs:
   - Manual intervention is required to approve the request.
   - The end host is configured to automatically request a certificate from the CA. Thus, operator intervention is no longer required at the time the enrollment request is sent to the CA server.

**Note**

If you configure the end host to automatically request certificates from the CA, you should have an additional authorization mechanism.
After the request is approved, the CA signs the request with its private key and returns the completed certificate to the end host.

The end host writes the certificate to a storage area such as NVRAM.

Certificate Enrollment Via Secure Device Provisioning

Secure Device Provisioning (SDP) is a web-based certificate enrollment interface that can be used to easily deploy PKI between two end devices, such as a Cisco IOS client and a Cisco IOS certificate server. SDP (also referred to as Trusted Transitive Introduction [TTI]) is a communication protocol that provides a bidirectional introduction between two end entities, such as a new network device and a Virtual Private Network (VPN). SDP involves the following three entities:

• Introducer--A mutually trusted device that introduces the petitioner to the registrar. The introducer can be a device user, such as a system administrator.
• Petitioner--A new device that is joined to the secure domain.
• Registrar--A certificate server or other server that authorizes the petitioner.

SDP is implemented over a web browser in three phases--welcome, introduction, and completion. Each phase is shown to the user via a web page. For more information on each phase and how SDP works, see the “Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI” module.

Certificate Revocation Why It Occurs

After each participant has successfully enrolled in the PKI, the peers are ready to begin negotiations for a secure connection with each other. Thus, the peers present their certificates for validation followed by a revocation check. After the peer verifies that the other peer’s certificate was issued by an authenticated CA, the CRL or Online Certificate Status Protocol (OCSP) server is checked to ensure that the certificate has not been revoked by the issuing CA. The certificate usually contains a certificate distribution point (CDP) in the form of a URL. Cisco IOS software uses the CDP to locate and retrieve the CRL. If the CDP server does not respond, the Cisco IOS software reports an error, which may result in the peer’s certificate being rejected.

Planning for a PKI

Planning for a PKI requires evaluating the requirements and expected use for each of the PKI components shown in Planning for a PKI, on page 5. It is recommended that you (or the network administrator) thoroughly plan the PKI before beginning any PKI configuration.

Although there are a number of approaches to consider when planning the PKI, this document begins with peer-to-peer communication and proceeds as shown in Planning for a PKI, on page 5. However you or the network administrator choose to plan the PKI, understand that certain decisions influence other decisions within the PKI. For example, the enrollment and deployment strategy could influence the planned CA hierarchy. Thus, it is important to understand how each component functions within the PKI and how certain component options are dependent upon decisions made earlier in the planning process.
Where to Go Next

As suggested in *Where to Go Next*, on page 6, you begin to configure a PKI by setting up and deploying RSA keys. For more information, see the module “Deploying RSA Keys Within a PKI.”

Additional References

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples</td>
<td>Cisco IOS Security Command Reference</td>
</tr>
<tr>
<td>Certificate enrollment: supported methods, enrollment profiles, configuration tasks</td>
<td>Configuring Certificate Enrollment for a PKI module in the <em>Cisco IOS Security Configuration Guide: Secure Connectivity</em></td>
</tr>
<tr>
<td>Certificate revocation and authorization: configuration tasks</td>
<td>Configuring Revocation and Authorization of Certificates in a PKI module in the <em>Cisco IOS Security Configuration Guide: Secure Connectivity</em></td>
</tr>
<tr>
<td>Cisco IOS certificate server overview information and configuration tasks</td>
<td>Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment module in the <em>Cisco IOS Security Configuration Guide: Secure Connectivity</em></td>
</tr>
<tr>
<td>Secure Device Provisioning: functionality overview and configuration tasks</td>
<td>Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI module in the <em>Cisco IOS Security Configuration Guide: Secure Connectivity</em></td>
</tr>
<tr>
<td>Storing RSA keys and certificates on a USB eToken</td>
<td>Storing PKI Credentials module in the <em>Cisco IOS Security Configuration Guide: Secure Connectivity</em></td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>--</td>
</tr>
</tbody>
</table>
### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
</table>
## Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

## Glossary

**CDP** -- certificate distribution point. Field within a digital certificate containing information that describes how to retrieve the CRL for the certificate. The most common CDPs are HTTP and LDAP URLs. A CDP may also contain other types of URLs or an LDAP directory specification. Each CDP contains one URL or directory specification.

**certificates** -- Electronic documents that bind a user’s or device’s name to its public key. Certificates are commonly used to validate a digital signature.

**CRL** -- certificate revocation list. Electronic document that contains a list of revoked certificates. The CRL is created and digitally signed by the CA that originally issued the certificates. The CRL contains dates for when the certificate was issued and when it expires. A new CRL is issued when the current CRL expires.

**CA** -- certification authority. Service responsible for managing certificate requests and issuing certificates to participating IPSec network devices. This service provides centralized key management for the participating devices and is explicitly trusted by the receiver to validate identities and to create digital certificates.

**peer certificate** -- Certificate presented by a peer, which contains the peer’s public key and is signed by the trustpoint CA.

**PKI** -- public key infrastructure. System that manages encryption keys and identity information for components of a network that participate in secured communications.

**RA** -- registration authority. Server that acts as a proxy for the CA so that CA functions can continue when the CA is offline. Although the RA is often part of the CA server, the RA could also be an additional application, requiring an additional device to run it.

**RSA keys** -- Public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router.
Deploying RSA Keys Within a PKI

This module explains how to set up and deploy Rivest, Shamir, and Adelman (RSA) keys within a public key infrastructure (PKI). An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router; that is, the end host must generate a pair of RSA keys and exchange the public key with the certification authority (CA) to obtain a certificate and enroll in a PKI.

Note

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

- Finding Feature Information, page 9
- Prerequisites for Configuring RSA Keys for a PKI, page 10
- Information About RSA Keys Configuration, page 10
- How to Set Up and Deploy RSA Keys Within a PKI, page 12
- Configuration Examples for RSA Key Pair Deployment, page 26
- Where to Go Next, page 30
- Additional References, page 30
- Feature Information for RSA Keys Within a PKI, page 31

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.
Prerequisites for Configuring RSA Keys for a PKI

• Before setting up and deploying RSA keys for a PKI, you should be familiar with the module Cisco IOS PKI Overview: Understanding and Planning a PKI.

• As of Cisco IOS Release 12.3(7)T, all commands that begin as “crypto ca” have been changed to begin as “crypto pki.” Although the router will still accept crypto ca commands, all output will be read back as crypto pki.

Information About RSA Keys Configuration

RSA Keys Overview

An RSA key pair consists of a public key and a private key. When setting up your PKI, you must include the public key in the certificate enrollment request. After the certificate has been granted, the public key will be included in the certificate so that peers can use it to encrypt data that is sent to the router. The private key is kept on the router and used both to decrypt the data sent by peers and to digitally sign transactions when negotiating with peers.

RSA key pairs contain a key modulus value. The modulus determines the size of the RSA key. The larger the modulus, the more secure the RSA key. However, keys with large modulus values take longer to generate, and encryption and decryption operations take longer with larger keys.

Note

As of Cisco IOS Release 12.4(11)T, peer public RSA key modulus values up to 4096 bits are automatically supported. The largest private RSA key modulus is 4096 bits. Therefore, the largest RSA private key a router may generate or import is 4096 bits. However, RFC 2409 restricts the private key size to 2048 bits or less for RSA encryption. The recommended modulus value for a CA is 2048 bits; the recommended modulus value for a client is also 2048 bits.

Usage RSA Keys Versus General-Purpose RSA Keys

There are two mutually exclusive types of RSA key pairs—usage keys and general-purpose keys. When you generate RSA key pairs (via the `crypto key generate rsa` command), you will be prompted to select either usage keys or general-purpose keys.

Usage RSA Keys

Usage keys consist of two RSA key pairs—one RSA key pair is generated and used for encryption and one RSA key pair is generated and used for signatures. With usage keys, each key is not unnecessarily exposed. (Without usage keys, one key is used for both authentication methods, increasing the exposure of that key.)

General-Purpose RSA Keys

General-purpose keys consist of only one RSA key pair that used for both encryption and signatures. General-purpose key pairs are used more frequently than usage key pairs.
How RSA Key Pairs are Associated with a Trustpoint

A trustpoint, also known as the certificate authority (CA), manages certificate requests and issues certificates to participating network devices. These services provide centralized key management for the participating devices and are explicitly trusted by the receiver to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

Reasons to Store Multiple RSA Keys on a Router

Configuring multiple RSA key pairs allows the Cisco IOS software to maintain a different key pair for each CA with which it is dealing or the software can maintain multiple key pairs and certificates with the same CA. As a result, the Cisco IOS software can match policy requirements for each CA without compromising the requirements specified by the other CAs, such as key length, key lifetime, and general-purpose versus usage keys.

Named key pairs (which are specified via the label key-label option) allow you to have multiple RSA key pairs, enabling the Cisco IOS software to maintain a different key pair for each identity certificate.

Benefits of Exportable RSA Keys

Caution

Exportable RSA keys should be carefully evaluated before use because using exportable RSA keys introduces the risk that these keys might be exposed. Any existing RSA keys are not exportable. New keys are generated as nonexportable by default. It is not possible to convert an existing nonexportable key to an exportable key.

As of Cisco IOS Release 12.2(15)T, users can share the private RSA key pair of a router with standby routers, therefore transferring the security credentials between networking devices. The key pair that is shared between two routers will allow one router to immediately and transparently take over the functionality of the other router. If the main router were to fail, the standby router could be dropped into the network to replace the failed router without the need to regenerate keys, reenroll with the CA, or manually redistribute keys.

Exporting and importing an RSA key pair also enables users to place the same RSA key pair on multiple routers so that all management stations using Secure Shell (SSH) can be configured with a single public RSA key.

Exportable RSA Keys in PEM-Formatted Files

Using privacy-enhanced mail (PEM)-formatted files to import or export RSA keys can be helpful for customers who are running Cisco IOS software Release 12.3(4)T or later and who are using secure socket layer (SSL) or secure shell (SSH) applications to manually generate RSA key pairs and import the keys back into their PKI applications. PEM-formatted files allow customers to directly use existing RSA key pairs on their Cisco IOS routers instead of generating new keys.
Passphrase Protection While Importing and Exporting RSA Keys

You have to include a passphrase to encrypt the PKCS12 file or the PEM file that will be exported, and when the PKCS12 or PEM file is imported, the same passphrase has to be entered to decrypt it. Encrypting the PKCS12 or PEM file when it is being exported, deleted, or imported protects the file from unauthorized access and use while it is being transported or stored on an external device.

The passphrase can be any phrase that is at least eight characters in length; it can include spaces and punctuation, excluding the question mark (?), which has special meaning to the Cisco IOS parser.

How to Convert an Exportable RSA Key Pair to a Nonexportable RSA Key Pair

Passphrase protection protects the external PKCS12 or PEM file from unauthorized access and use. To prevent an RSA key pair from being exported, it must be labeled "nonexportable." To convert an exportable RSA key pair into a nonexportable key pair, the key pair must be exported and then reimported without specifying the "exportable" keyword.

How to Set Up and Deploy RSA Keys Within a PKI

Generating an RSA Key Pair

Perform this task to manually generate an RSA key pair.

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto key generate rsa [general-keys | usage-keys | signature | encryption] [label key-label] [exportable] [modulus modulus-size] [storage devicename:] [on devicename:]
4. exit
5. show crypto key mypubkey rsa

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# configure terminal</td>
<td></td>
</tr>
</tbody>
</table>
### Generating an RSA Key Pair

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td>(Optional) Generates the RSA key pair for the certificate server.</td>
</tr>
<tr>
<td>crypto key generate rsa [general-keys</td>
<td>The storage keyword specifies the key storage location.</td>
</tr>
<tr>
<td>usage-keys</td>
<td>• When specifying a label name by specifying the key-label argument, signature</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>If the exportable RSA key pair is manually generated after the CA certificate</td>
</tr>
<tr>
<td></td>
<td>has been generated, and before issuing the no shutdown command, then use</td>
</tr>
<tr>
<td></td>
<td>the certificate server certificate and the private key.</td>
</tr>
<tr>
<td></td>
<td>• The on keyword specifies that the RSA key pair is created on the</td>
</tr>
<tr>
<td></td>
<td>specified device, including a Universal Serial Bus (USB) token, local</td>
</tr>
<tr>
<td></td>
<td>disk, or NVRAM. The name of the device is followed by a colon (:).</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>exits global configuration mode.</td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>(Optional) Displays the RSA public keys of your router.</td>
</tr>
<tr>
<td>show crypto key mypubkey rsa</td>
<td>This step allows you to verify that the RSA key pair has been successfully</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>generated.</td>
</tr>
<tr>
<td>Router# show crypto key mypubkey rsa</td>
<td></td>
</tr>
</tbody>
</table>

### What to Do Next

After you have successfully generated an RSA key pair, you can proceed to any of the additional tasks in this module to generate additional RSA key pairs, perform export and import of RSA key pairs, or configure additional security parameters for the RSA key pair (such as encrypting or locking the private key).
Managing RSA Key Pairs and Trustpoint Certificates

Perform this task to configure the router to generate and store multiple RSA key pairs, associate the key pairs with a trustpoint, and get the certificates for the router from the trustpoint.

Before You Begin

You must have already generated an RSA key pair as shown in the task “Generating an RSA Key Pair task.”

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. rsakeypair key-label [key-size [encryption-key-size]]
5. enrollment selfsigned
6. subject-alt-name name
7. exit
8. crypto pki enroll name
9. exit
10. show crypto key mypubkey rsa

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto pki trustpoint name</td>
<td>Creates a trustpoint and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# crypto pki trustpoint TESTCA</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> rsakeypair key-label [key-size [encryption-key-size]]</td>
<td>(Optional) The key-label argument specifies the name of the RSA key pair generated during enrollment (if it does not already exist or if the auto-enroll regenerate command is configured) to be used with the</td>
</tr>
</tbody>
</table>
### Deploying RSA Keys Within a PKI

#### Managing RSA Key Pairs and Trustpoint Certificates

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>trustpoint certificate. By default, the fully qualified domain name (FQDN) key is used.</td>
</tr>
</tbody>
</table>
| Router(ca-trustpoint)# rsakeypair fancy-keys | • (Optional) The *key-size* argument specifies the size of the RSA key pair. The recommended key size is 2048 bits.  
• (Optional) The *encryption-key-size* argument specifies the size of the second key, which is used to request separate encryption, signature keys, and certificates. |

**Step 5** **enrollment selfsigned**  
(Optional) Specifies self-signed enrollment for a trustpoint.  
**Example:**  
Router(ca-trustpoint)# enrollment selfsigned

**Step 6** **subject-alt-name name**  
(Optional) The *name* argument specifies the trustpoint’s name in the Subject Alternative Name (subjectAltName) field in the X.509 certificate, which is contained in the trustpoint certificate. By default, the Subject Alternative Name field is not included in the certificate.  
**Example:**  
Router(ca-trustpoint)# subject-alt-name TESTCA  
**Note** This X.509 certificate field is defined in RFC 2511.  
This option is used to create a self-signed trustpoint certificate for the router that contains the trustpoint name in the Subject Alternative Name (subjectAltName) field. This Subject Alternative Name can be used only when the enrollment selfsigned command is specified for self-signed enrollment in the trustpoint policy.

**Step 7** **exit**  
Exits ca-trustpoint configuration mode.  
**Example:**  
Router(ca-trustpoint)# exit

**Step 8** **crypto pki enroll name**  
Requests the certificates for the router from the trustpoint.  
The *name* argument specifies the trustpoint name. Once this command is entered, answer the prompts.  
**Example:**  
Router(config)# crypto pki enroll TESTCA  
**Note** Use the same trustpoint name entered with the crypto pki trustpoint command.  
% Include the router serial number in the subject name? [yes/no]: no
### Purpose

**Command or Action**

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Include an IP address in the subject name? [no]:</td>
</tr>
<tr>
<td>Generate Self Signed Router Certificate? [yes/no]: yes</td>
</tr>
<tr>
<td>Router Self Signed Certificate successfully created</td>
</tr>
<tr>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td>(Optional) Displays the RSA public keys of your router.</td>
</tr>
<tr>
<td>This step allows you to verify that the RSA key pair has been successfully generated.</td>
</tr>
</tbody>
</table>

---

**Example**

The following example shows how to create a self-signed trustpoint certificate for the router that contains the trustpoint name in the Subject Alternative Name (subjectAltName) field:

```
Router> enable
Router# configure terminal
Router(config)#crypto pki trustpoint TESTCA
Router(ca-trustpoint)#hash sha256
Router(ca-trustpoint)#rsakeypair testca-rsa-key 2048
Router(ca-trustpoint)#exit
Router(config)#crypto pki enroll TESTCA
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Generate Self Signed Router Certificate? [yes/no]: yes
Router Self Signed Certificate successfully created
Router(config)#
Router(config)#exit
Router#
```

The following certificate is created:

```
Router#show crypto pki certificate verbose
Router Self-Signed Certificate
Status: Available
Version: 3
```
Exporting and Importing RSA Keys

This section contains the following tasks that can be used for exporting and importing RSA keys. Whether you are using PKCS12 files or PEM files, exportable RSA keys allow you to use existing RSA keys on Cisco IOS routers instead of having to generate new RSA keys if the main router were to fail.

Exporting and Importing RSA Keys in PKCS12 Files

Exporting and importing RSA key pairs enables users to transfer security credentials between devices. The key pair that is shared between two devices allows one device to immediately and transparently take over the functionality of the other router.

Before You Begin

You must generate an RSA key pair and mark it "exportable" as specified in the “Generating an RSA Key Pair” task.
**Note**

- You cannot export RSA keys that existed on the router before your system was upgraded to Cisco IOS Release 12.2(15)T or later. You have to generate new RSA keys and label them as "exportable" after you upgrade the Cisco IOS software.

- When you import a PKCS12 file that was generated by a third-party application, the PKCS12 file must include a CA certificate.

- If you want reexport an RSA key pair after you have already exported the key pair and imported them to a target router, you must specify the `exportable` keyword when you are importing the RSA key pair.

- The largest RSA key a router may import is 2048-bits.

**SUMMARY STEPS**

1. `crypto pki trustpoint name`
2. `rsakeypair key-label [key-size [encryption-key-size]]`
3. `exit`
4. `crypto pki export trustpointname pkcs12 destination-url password password-phrase`
5. `crypto pki import trustpointname pkcs12 source-url password password-phrase`
6. `exit`
7. `show crypto key mypubkey rsa`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>crypto pki trustpoint <em>name</em></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# crypto pki trustpoint my-ca</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>rsakeypair <em>key-label</em> [key-size [encryption-key-size]]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(ca-trustpoint)# rsakeypair my-keys</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(ca-trustpoint)# exit</td>
</tr>
</tbody>
</table>
### Purpose

**Step 4**

**crypto pki export trustpointname pkcs12 destination-url password password-phrase**

**Example:**

Router(config)# crypto pki export my-ca pkcs12 tftp://tftpserver/my-keys password mypassword123

Exports the RSA keys through the trustpoint name.

- The `trustpointname` argument enters the name of the trustpoint that issues the certificate that a user is going to export. When exporting the PKCS12 file, the trustpoint name is the RSA key name.
- The `destination-url` argument enters the file system location of the PKCS12 file to which a user wants to import the RSA key pair. See the `crypto pki export pkcs12 password` command page for more information.
- The `password-phrase` argument must be entered to encrypt the PKCS12 file for export.

**Step 5**

**crypto pki import trustpointname pkcs12 source-url password password-phrase**

**Example:**

Router(config)# crypto pki import my-ca pkcs12 tftp://tftpserver/my-keys password mypassword123

Imports the RSA keys to the target router.

- The `trustpointname` argument enters the name of the trustpoint that issues the certificate that a user is going to export or import. When importing, the trustpoint becomes the RSA key name.
- The `source-url` argument specifies the file system location of the PKCS12 file to which a user wants to export the RSA key pair. See the `crypto pki import pkcs12 password` command page for more information.
- The `password-phrase` must be entered to undo encryption when the RSA keys are imported.

**Step 6**

**exit**

**Example:**

Router(config)# exit

Exits global configuration mode.

**Step 7**

**show crypto key mypubkey rsa**

**Example:**

Router# show crypto key mypubkey rsa

(Optional) Displays the RSA public keys of your router.

---

### Exporting and Importing RSA Keys in PEM-Formatted Files

Perform this task to export or import RSA key pairs in PEM files.

**Before You Begin**

You must generate an RSA key pair and mark it "exportable" as specified the “Generating an RSA Key Pair” task.
**Summary Steps**

1. `crypto key generate rsa {usage-keys | general-keys} label key-label [exportable]`
2. `crypto pki export trustpoint pem {terminal | url destination-url} {3des | des} password password-phrase`
3. `crypto pki import trustpoint pem [check | exportable | usage-keys] {terminal | url source-url} password password-phrase`
4. exit
5. `show crypto key mypubkey rsa`

**Detailed Steps**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Generates the RSA key pair.</td>
</tr>
<tr>
<td>`crypto key generate rsa {usage-keys</td>
<td>general-keys} label key-label [exportable]`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# crypto key generate rsa general-keys label mykey exportable</code></td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2** | Exports the certificates and RSA keys that are associated with a trustpoint in a PEM-formatted file. |
| `crypto pki export trustpoint pem {terminal | url destination-url} {3des | des} password password-phrase` | |
| **Example:** | |
| `Router(config)# crypto pki export mycs pem url nvram: 3des password mypassword123` | |

- Enter the trustpoint name that is associated with the exported certificate and RSA key pair. The trustpoint name must match the name that was specified through the `crypto pki trustpoint` command.
- Use the terminal keyword to specify the certificate and RSA key pair that is displayed in PEM format on the console terminal.
- Use the url keyword and destination-url argument to specify the URL of the file system where your router should export the certificates and RSA key pair.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td>Imports certificates and RSA keys to a trustpoint from PEM-formatted files.</td>
</tr>
<tr>
<td>crypto pki import trustpoint pem [check</td>
<td>exportable</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# crypto pki import mycs2 pem url nvram: password mypassword123</td>
</tr>
<tr>
<td></td>
<td>• Enter the trustpoint name that is associated with the imported certificate and RSA key pair. The trustpoint name must match the name that was specified through the crypto pki trustpoint command.</td>
</tr>
<tr>
<td></td>
<td>• (Optional) Use the check keyword to specify that an outdated certificate is not allowed.</td>
</tr>
<tr>
<td></td>
<td>• (Optional) Use the exportable keyword to specify that the imported RSA key pair can be exported again to another Cisco device such as a router.</td>
</tr>
<tr>
<td></td>
<td>• (Optional) Use the usage-keys argument to specify that two RSA special usage key pairs will be imported (that is, one encryption pair and one signature pair), instead of one general-purpose key pair.</td>
</tr>
<tr>
<td></td>
<td>• Use the source-url argument to specify the URL of the file system where your router should import the certificates and RSA key pairs.</td>
</tr>
<tr>
<td></td>
<td>• Use the password-phrase argument to specify the encrypted password phrase that is used to encrypt the PEM file for import.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The password phrase can be any phrase that is at least eight characters in length; it can include spaces and punctuation, excluding the question mark (?), which has special meaning to the Cisco IOS parser.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>If you do not want the key to be exportable from your CA, import it back to the CA after it has been exported as a nonexportable key pair. Thus, the key cannot be taken off again.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# exit</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>(Optional) Displays the RSA public keys of your router.</td>
</tr>
<tr>
<td>show crypto key mypubkey rsa</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Router# show crypto key mypubkey rsa</td>
</tr>
</tbody>
</table>
Encrypting and Locking Private Keys on a Router

Digital signatures are used to authenticate one device to another device. To use digital signatures, private information (the private key) must be stored on the device that is providing the signature. The stored private information may aid an attacker who steals the hardware device that contains the private key; for example, a thief might be able to use the stolen router to initiate a secure connection to another site by using the RSA private keys stored in the router.

**Note**

RSA keys are lost during password recovery operations. If you lose your password, the RSA keys will be deleted when you perform the password recovery operation. (This function prevents an attacker from performing password recovery and then using the keys.)

To protect the private RSA key from an attacker, a user can encrypt the private key that is stored in NVRAM via a passphrase. Users can also “lock” the private key, which blocks new connection attempts from a running router and protects the key in the router if the router is stolen by an attempted attacker.

Perform this task to encrypt and lock the private key that is saved to NVRAM.

**Before You Begin**

Before encrypting or locking a private key, you should perform the following tasks:

- Generate an RSA key pair as shown in the task “Generating an RSA Key Pair, on page 12.”
- Optionally, you can authenticate and enroll each router with the CA server.

**Note**

**Backward Compatibility Restriction**

Any image prior to Cisco IOS Release 12.3(7)T does not support encrypted keys. To prevent your router from losing all encrypted keys, ensure that only unencrypted keys are written to NVRAM before booting an image prior to Cisco IOS Release 12.3(7)T.

If you must download an image prior to Cisco IOS Release 12.3(7)T, decrypt the key and immediately save the configuration so the downloaded image does not overwrite the configuration.

**Interaction with Applications**

An encrypted key is not effective after the router boots up until you manually unlock the key (via the `crypto key unlock rsa` command). Depending on which key pairs are encrypted, this functionality may adversely affect applications such as IP security (IPsec), SSH, and SSL; that is, management of the router over a secure channel may not be possible until the necessary key pair is unlocked.
SUMMARY STEPS

1. crypto key encrypt [write] rsa [name key-name] passphrase passphrase
2. exit
3. show crypto key mypubkey rsa
4. crypto key lock rsa name key-name passphrase passphrase
5. show crypto key mypubkey rsa
6. crypto key unlock rsa [name key-name] passphrase passphrase
7. configure terminal
8. crypto key decrypt [write] rsa [name key-name ] passphrase passphrase

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>crypto key encrypt [write] rsa [name key-name] passphrase passphrase</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# crypto key encrypt write rsa name pki.example.com passphrase password</td>
</tr>
<tr>
<td></td>
<td>Encrypts the RSA keys. After this command is issued, the router can continue to use the key; the key remains unlocked.</td>
</tr>
<tr>
<td></td>
<td>If the write keyword is not issued, the configuration must be manually written to NVRAM; otherwise, the encrypted key will be lost next time the router is reloaded.</td>
</tr>
</tbody>
</table>

| **Step 2** | exit |
| **Example:** | Router(config)# exit |
| | Exits global configuration mode. |

| **Step 3** | show crypto key mypubkey rsa |
| **Example:** | Router# show crypto key mypubkey rsa |
| | (Optional) Shows that the private key is encrypted (protected) and unlocked. |
| | You can also use this command to verify that applications such as Internet Key Exchange (IKE) and SSH are properly working after the key has been encrypted. |

| **Step 4** | crypto key lock rsa name key-name ] passphrase passphrase |
| **Example:** | Router# crypto key lock rsa name pki.example.com passphrase password |
| | (Optional) Locks the encrypted private key on a running router. |
| | After the key is locked, it cannot be used to authenticate the router to a peer device. This behavior disables any IPSec or SSL connections that use the locked key. Any existing IPSec tunnels created on the basis of the locked key will be closed. If all RSA keys are locked, SSH will automatically be disabled. |

| **Step 5** | show crypto key mypubkey rsa |
| **Example:** | Router# show crypto key mypubkey rsa |
| | (Optional) Shows that the private key is protected and locked. The output will also show failed connection attempts via applications such as IKE, SSH, and SSL. |
### Removing RSA Key Pair Settings

An RSA key pair may need to be removed for one of the following reasons:

- During manual PKI operations and maintenance, old RSA keys can be removed and replaced with new keys.
- An existing CA is replaced and the new CA requires newly generated keys; for example, the required key size might have changed in an organization so you would have to delete the old 1024-bit keys and generate new 2048-bit keys.
- The peer router's public keys can be deleted in order to help debug signature verification problems in IKEv1 and IKEv2. Keys are cached by default with the lifetime of the certificate revocation list (CRL) associated with the trustpoint.

Perform this task to remove all RSA keys or the specified RSA key pair that has been generated by your router.
### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `crypto key zeroize rsa [key-pair-label]`
4. `crypto key zeroize pubkey-chain [index]`
5. `exit`
6. `show crypto key mypubkey rsa`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td><strong>Example:</strong> Router&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td><strong>Example:</strong> Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Deletes RSA key pairs from your router.</td>
</tr>
<tr>
<td><code>crypto key zeroize rsa [key-pair-label]</code></td>
<td><strong>Example:</strong> Router(config)# crypto key zeroize rsa fancy-keys</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Deletes the remote peer’s public key from the cache.</td>
</tr>
<tr>
<td><code>crypto key zeroize pubkey-chain [index]</code></td>
<td><strong>Example:</strong> Router(config)# crypto key zeroize pubkey-chain</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td><code>exit</code></td>
<td><strong>Example:</strong> Router(config)# exit</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.</td>
</tr>
<tr>
<td><code>show crypto key mypubkey rsa</code></td>
<td><strong>Example:</strong> Router# show crypto key mypubkey rsa</td>
</tr>
</tbody>
</table>
Configuration Examples for RSA Key Pair Deployment

Generating and Specifying RSA Keys Example

The following example is a sample trustpoint configuration that shows how to generate and specify the RSA key pair "exampleCAkeys":

```
crypto key generate rsa general-purpose exampleCAkeys
crypto ca trustpoint exampleCAkeys
   enroll url http://exampleCAkeys/certsrv/mscep/mscep.dll
rsakeypair exampleCAkeys 1024 1024
```

Exporting and Importing RSA Keys Examples

Exporting and Importing RSA Keys in PKCS12 Files Example

In the following example, an RSA key pair "mynewkp" is generated on Router A, and a trustpoint name "mynewtp" is created and associated with the RSA key pair. The trustpoint is exported to a TFTP server, so that it can be imported on Router B. By importing the trustpoint "mynewtp" to Router B, the user has imported the RSA key pair "mynewkp" to Router B.

**Router A**

```
crypto key generate rsa general label mykeys exportable
   ! The name for the keys will be:mynewkp
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.
   How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys ...[OK]
   !
crypto pki trustpoint mynewtp
rsakeypair mykeys
exit
crypto pki export mytp pkcs12 flash:myexport password mypassword123
   Destination filename [myexport]? Writing pkcs12 file to tftp:/mytftpserver/myexport
CRYPTO_PKI:Exported PKCS12 file successfully.
   Verifying checksum... OK (0x3307)
   !
   July 8 17:30:09 GMT:%CRYPTO-6-PKCS12EXPORT_SUCCESS:PKCS #12 Successfully Exported.
```

**Router B**

```
crypto pki import mynewtp pkcs12 flash:myexport password mypassword123
   Source filename [myexport]? CRYPTO_PKI:Imported PKCS12 file successfully.
   !
   July 8 18:07:50 GMT:%CRYPTO-6-PKCS12IMPORT_SUCCESS:PKCS #12 Successfully Imported.
```
Exporting and Importing and RSA Keys in PEM Files Example

The following example shows the generation, exportation, and importation fo the RSA key pair "mytp", and verifies its status:

! Generate the key pair
!
Router(config)# crypto key generate rsa general-purpose label mytp exportable

The name for the keys will be: mytp

Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [512]: 2048

% Generating 2048 bit RSA keys ...[OK]

! Archive the key pair to a remote location, and use a good password.
!
Router(config)# crypto pki export mytp pem url nvram:mytp 3des password mypassword123

% Key name:mytp
Usage:General Purpose Key
Exporting public key...
Destination filename [mytp.pub]?
Writing file to nvram:mytp.pub
Exporting private key...
Destination filename [mytp.prv]?
Writing file to nvram:mytp.prv

! Import the key as a different name.
!
Router(config)# crypto pki import mytp2 pem url nvram:mytp2 password mypassword123

% Importing public key or certificate PEM file...
Source filename [mytp2.pub]?
Reading file from nvram:mytp2.pub
% Importing private key PEM file...
Source filename [mytp2.prv]?
Reading file from nvram:mytp2.prv% Key pair import succeeded.
!
! After the key has been imported, it is no longer exportable.
!
! Verify the status of the key.
!
Router# show crypto key mypubkey rsa

% Key pair was generated at:18:04:56 GMT Jun 6 2011
Key name:mycs
Usage:General Purpose Key
Key is exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253 9C30C12E 2959B73F B1DF9FAD 86F88192 7D4FA4D2 8A7FB49 9045BA9B 373A31CB A6B1B8F4 329F2E7E 8A50997E AABDFEPA 23C29E19 C45F4F05 DBB2FA51 4B7E9F7F A1095115 759D8BC3 5DFB5D7F BCF655BF 6371DB12 A8287795 78BC6D8A D31B2486 C9C96D2C 270B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001
% Key pair was generated at:18:17:25 GMT Jun 6 2011
Key name:mycs2
Usage:General Purpose Key
Key is not exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253 9C30C12E 2959B73F B1DF9FAD 86F88192 7D4FA4D2 8A7FB49 9045BA9B 373A31CB A6B1B8F4 329F2E7E 8A50997E AABDFEPA 23C29E19 C45F4F05 DBB2FA51 4B7E9F7F A1095115 759D8BC3 5DFB5D7F BCF655BF 6371DB12 A8287795 78BC6D8A D31B2486 C9C96D2C 270B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001
Exporting Router RSA Key Pairs and Certificates from PEM Files Example

The following example shows how to generate and export the RSA key pair "aaa" and certificates of the router in PEM files that are associated with the trustpoint "mycs." This example also shows PEM-formatted files, which include PEM boundaries before and after the base64-encoded data, that are used by other SSL and SSH applications.

Router(config)# crypto key generate rsa general-keys label aaa exportable
The name for the keys will be: aaa
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [512]:
% Generating 512 bit RSA keys ... [OK]

Router(config)# crypto pki trustpoint mycs

Router(ca-trustpoint)# enrollment url http://mycs

Router(ca-trustpoint)# rsakeypair aaa

Router(ca-trustpoint)# exit

Router(config)# crypto pki authenticate mycs
Certificate has the following attributes:
Fingerprint:C21514AC 12815946 09F635ED FB6CF31
% Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.

Router(config)# crypto pki enroll mycs
% Start certificate enrollment..
% Create a challenge password. You will need to verbally provide this password to the CA Administrator in order to revoke your certificate. For security reasons your password will not be saved in the configuration. Please make a note of it.
Password:
Re-enter password:
% The fully-qualified domain name in the certificate will be: Router
% The subject name in the certificate will be: host.example.com
% Include the router serial number in the subject name? [yes/no]: n
% Include an IP address in the subject name? [no]: n
Request certificate from CA? [yes/no]: y
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the fingerprint.
Router(config)# Fingerprint:8DA777BC 08477073 A5BE2403 812DD157
00:29:11:%CRYPTO-6-CERTRET:Certificate received from Certificate Authority

Router(config)# crypto ca export aaa pem terminal 3des password
% CA certificate:
-----BEGIN CERTIFICATE-----
MIICczCCAgIBAgIBATANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJVUzES
<snip>
waDeNOSI3NdA0AwgAIBAgIBATANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJVUzES
-----END CERTIFICATE-----
% Key name: aaa
Usage: General Purpose Key
-----BEGIN RSA PRIVATE KEY-----
Proc-Type:4,ENCRYPTED
DEK-Info:DES-EDE3-CBC,0B1B2B3
 Urugu8nqOwCVQX5nbzzYHI2vGLUnpH/IXTsJuNjRVj8AAUpGK7VnPCT87
<snip>
 kLOCxKg72gMku9uUl1rL5nF5slzAtoC0zfU4-
Importing Router RSA Key Pairs and Certificate from PEM Files Example

The following example shows how to import the RSA key pairs and certificate to the trustpoint “ggg” from PEM files via TFTP:

```
Router(config)# crypto pki import ggg pem url tftp://10.1.1.2/username/msca password
```

Encrypting and Locking Private Keys on a Router Examples

Configuring and Verifying an Encrypted Key Example

The following example shows how to encrypt the RSA key “pki-123.example.com.” Thereafter, the `show crypto key mypubkey rsa` command is issued to verify that the RSA key is encrypted (protected) and unlocked.

```
Router(config)# crypto key encrypt rsa name pki-123.example.com passphrase password
Router(config)# exit
Router# show crypto key mypubkey rsa
% Key pair was generated at:00:15:32 GMT Jun 25 2003
Key name:pki-123.example.com
Usage:General Purpose Key
*** The key is protected and UNLOCKED. ***
Key is not exportable.
Key Data:
305C300D06092A86 4886F70D 01010105 00034B00 30480241 00E0CC9A 1D23B52C
CD00910C ABD392AE BA6D0E3F FC47A0EF 8AFE340 0EC1E62B D40E7DCC
```
Configuring and Verifying a Locked Key Example

The following example shows how to lock the key "pki-123.example.com." Thereafter, the show crypto key mypubkey rsa command is issued to verify that the key is protected (encrypted) and locked.

```
Router# crypto key lock rsa name pki-123.example.com passphrase password
!
Router# show crypto key mypubkey rsa

% Key pair was generated at: 20:29:41 GMT Jun 20 2003
Key name: pki-123.example.com
Usage: General Purpose Key
*** The key is protected and LOCKED. ***
Key is exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D7808D C5FF14AC
0D2B55AC 5D199F2F 7CB4B355 C555E07B 6D0DECBE 4519B1F0 75B12D6F 902D6E9F
B6FDAD8D 654EF851 5701D5D7 EDA047ED 9A2A619D 5639DF18 EB020301 0001
```

Where to Go Next

After you have generated an RSA key pair, you should set up the trustpoint. If you have already set up the trustpoint, you should authenticate and enroll the routers in a PKI. For information on enrollment, see the module "Configuring Certificate Enrollment for a PKI."

Additional References

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of PKI, including RSA keys, certificate enrollment, and CAs</td>
<td>Cisco IOS PKI Overview: Understanding and Planning a PKI</td>
</tr>
</tbody>
</table>
### Related Topic

<table>
<thead>
<tr>
<th>PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cisco IOS Security Command Reference</td>
</tr>
<tr>
<td>Recommended cryptographic algorithms</td>
<td>Next Generation Encryption</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 2409</td>
<td>The Internet Key Exchange (IKE)</td>
</tr>
<tr>
<td>RFC 2511</td>
<td>Internet X.509 Certificate Request Message Format</td>
</tr>
</tbody>
</table>

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

### Feature Information for RSA Keys Within a PKI

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

### Table 1: Feature Information for RSA Keys Within a PKI

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Software Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
</table>
| Cisco IOS 4096-Bit Public Key Support | 12.4(12)T         | This feature introduces Cisco IOS 4096-bit peer public key support. The following section provides information about this feature:  
  - RSA Keys Overview, on page 10 |
| Exporting and Importing RSA Keys    | 12.2(15)T         | This feature allows you to transfer security credentials between devices by exporting and importing RSA keys. The key pair that is shared between two devices will allow one device to immediately and transparently take over the functionality of the other router.  
  The following sections provide information about this feature:  
  - Benefits of Exportable RSA Keys, on page 11  
  - Exporting and Importing RSA Keys in PKCS12 Files, on page 17  
  The following commands were introduced or modified by this feature:  
  `crypto ca export pkcs12, crypto ca import pkcs12, crypto key generate rsa (IKE)` |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Software Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
</table>
| Import of RSA Key Pair and Certificates in PEM Format | 12.3(4)T          | This feature allows customers to use PEM-formatted files to import or export RSA key pairs. PEM-formatted files allow customers to directly use existing RSA key pairs on their Cisco IOS routers instead of generating new keys. The following sections provide information about this feature:  
  • Benefits of Exportable RSA Keys, on page 11  
  • Exporting and Importing RSA Keys in PEM-Formatted Files, on page 19  
  
The following commands were introduced by this feature: `crypto ca export pem`, `crypto ca import pem`, `crypto key export pem`, `crypto key import pem` |
| Multiple RSA Key Pair Support                         | 12.2(8)T          | This feature allows a user to configure a router to have multiple RSA key pairs. Thus, the Cisco IOS software can maintain a different key pair for each identity certificate. The following sections provide information about this feature:  
  • Reasons to Store Multiple RSA Keys on a Router, on page 11  
  • Managing RSA Key Pairs and Trustpoint Certificates, on page 14  
  
The following commands were introduced or modified by this feature: `crypto key generate rsa`, `crypto key zeroize rsa`, `rsa keypair` |
This feature allows a user to encrypt and lock the RSA private keys that are used on a Cisco IOS router, thereby, preventing unauthorized use of the private keys.

The following section provides information about this feature:

- **Encrypting and Locking Private Keys on a Router**

The following commands were introduced or modified by this feature: `crypto key decrypt rsa`, `crypto key encrypt rsa`, `crypto key lock rsa`, `crypto key unlock rsa`, `show crypto key mypubkey rsa`

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Software Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protected Private Key Storage</td>
<td>12.3(7)T</td>
<td>This feature allows a user to encrypt and lock the RSA private keys that are used on a Cisco IOS router, thereby, preventing unauthorized use of the private keys. The following section provides information about this feature: - Encrypting and Locking Private Keys on a Router</td>
</tr>
<tr>
<td>RSA 4096-bit Key Generation in Software Crypto Engine Support</td>
<td>15.1(1)T</td>
<td>The range value for the modulus keyword value for the <code>crypto key generate rsa</code> command is extended from 360 to 2048 bits to 360 to 4096 bits.</td>
</tr>
</tbody>
</table>
The IOS Performance Monitoring and Optimization feature provides a way to characterize the performance within the Public Key Infrastructure (PKI) subsystem and debug and analyze PKI performance related issues. This feature is discussed in further detail in the IOS Performance Monitoring and Optimization feature document.

This feature also includes the following enhancements that can be found in this document:

• A self-signed trustpoint certificate can be created for the router that contains the trustpoint name in the Subject Alternative Name (subjectAltName) field.

• A peer router’s public keys can be deleted to help debug signature verification problems in IKE version 1 and IKE version 2 and optimize the peer router’s performance as a result of taking this action.

These features can be found in the following sections:

The following commands were introduced or modified by this feature: `crypto key zeroize pubkey-chain, subject-alt-name`
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Software Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI IPv6 Support for VPN Solutions</td>
<td>15.2(1)T</td>
<td>The <code>crypto pki export pem</code> command was modified. Support was added in the CLI for hiding the password in an exported PEM-formatted file with the introduction of the <code>password</code> keyword followed by the <code>password-phrase</code> argument. The <code>crypto pki export pkcs12 password</code> command was modified. Support was added in the CLI for hiding the password in an exported PKCS12-formatted file with the introduction of the <code>password</code> keyword followed by the <code>password-phrase</code> argument. The <code>crypto pki import pem</code> command was modified. Support was added in the CLI for hiding the password in an imported PEM-formatted file with the introduction of the <code>password</code> keyword followed by the <code>password-phrase</code> argument. The <code>crypto pki import pkcs12 password</code> command was modified. Support was added in the CLI for hiding the password in an imported PKCS12-formatted file with the introduction of the <code>password</code> keyword followed by the <code>password-phrase</code> argument.</td>
</tr>
</tbody>
</table>
Configuring Authorization and Revocation of Certificates in a PKI

This module describes how to configure authorization and revocation of certificates in a public key infrastructure (PKI). It includes information on high-availability support for the certificate server.

Note
Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

- Finding Feature Information, page 37
- Prerequisites for Authorization and Revocation of Certificates, page 38
- Restrictions for Authorization and Revocation of Certificates, page 38
- Information About Authorization and Revocation of Certificates, page 38
- How to Configure Authorization and Revocation of Certificates for Your PKI, page 46
- Configuration Examples for Setting Up Authorization and Revocation of Certificates, page 70
- Additional References, page 83
- Feature Information for Certificate Authorization and Revocation, page 83

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.
Prerequisites for Authorization and Revocation of Certificates

Plan Your PKI Strategy

Tip

It is strongly recommended that you plan your entire PKI strategy before you begin to deploy actual certificates.

Authorization and revocation can occur only after you or a network administrator have completed the following tasks:

- Configured the certificate authority (CA).
- Enrolled peer devices with the CA.
- Identified and configured the protocol (such as IP Security [IPsec] or secure socket layer [SSL]) that is to be used for peer-to-peer communication.

You should decide which authorization and revocation strategy you are going to configure before enrolling peer devices because the peer device certificates might have to contain authorization and revocation-specific information.

"crypto ca" to "crypto pki" CLI Change

As of Cisco IOS Release 12.3(7)T, all commands that begin as "crypto ca" have been changed to begin as "crypto pki." Although the router will still accept crypto ca commands, all output will be read back as crypto pki.

High Availability

For high availability, IPsec-secured Stream Control Transmission Protocol (SCTP) must be configured on both the active and the standby routers. For synchronization to work, the redundancy mode on the certificate servers must be set to ACTIVE/STANDBY after you configure SCTP.

Restrictions for Authorization and Revocation of Certificates

PKI High Availability (HA) support of intra-chassis stateful switchover (SSO) redundancy is currently not supported on all switches running the Cisco IOS Release 12.2 S software. See Cisco bug CSCtb59872 for more information.

Information About Authorization and Revocation of Certificates

PKI Authorization

PKI authentication does not provide authorization. Current solutions for authorization are specific to the router that is being configured, although a centrally managed solution is often required.
There is not a standard mechanism by which certificates are defined as authorized for some tasks and not for others. This authorization information can be captured in the certificate itself if the application is aware of the certificate-based authorization information. But this solution does not provide a simple mechanism for real-time updates to the authorization information and forces each application to be aware of the specific authorization information embedded in the certificate.

When the certificate-based ACL mechanism is configured as part of the trustpoint authentication, the application is no longer responsible for determining this authorization information, and it is no longer possible to specify for which application the certificate is authorized. In some cases, the certificate-based ACL on the router gets so large that it cannot be managed. Additionally, it is beneficial to retrieve certificate-based ACL indications from an external server. (For more information on using certificate-based ACLs for authentication, see the section "When to Use Certificate-Based ACLs for Authorization or Revocation, on page 43.")

Current solutions to the real-time authorization problem involve specifying a new protocol and building a new server (with associated tasks, such as management and data distribution).

**PKI and AAA Server Integration for Certificate Status**

Integrating your PKI with an authentication, authorization, and accounting (AAA) server provides an alternative online certificate status solution that leverages the existing AAA infrastructure. Certificates can be listed in the AAA database with appropriate levels of authorization. For components that do not explicitly support PKI-AAA, a default label of "all" from the AAA server provides authorization. Likewise, a label of "none" from the AAA database indicates that the specified certificate is not valid. (The absence of any application label is equivalent, but "none" is included for completeness and clarity). If the application component does support PKI-AAA, the component may be specified directly; for example, the application component could be “ipsec,” “ssl,” or “osp.” (ipsec=IP Security, ssl=Secure Sockets Layer, and osp=Open Settlement Protocol.)

**Note**

Currently, no application component supports specification of the application label.

- There may be a time delay when accessing the AAA server. If the AAA server is not available, the authorization fails.

**RADIUS or TACACS+ Choosing a AAA Server Protocol**

The AAA server can be configured to work with either the RADIUS or TACACS+ protocol. When you are configuring the AAA server for the PKI integration, you must set the RADIUS or TACACS attributes that are required for authorization.

If the RADIUS protocol is used, the password that is configured for the username in the AAA server should be set to “cisco,” which is acceptable because the certificate validation provides authentication and the AAA database is only being used for authorization. When the TACACS protocol is used, the password that is configured for the username in the AAA server is irrelevant because TACACS supports authorization without requiring authentication (the password is used for authentication).

In addition, if you are using TACACS, you must add a PKI service to the AAA server. The custom attribute “cert-application=all” is added under the PKI service for the particular user or usergroup to authorize the specific username.
Attribute-Value Pairs for PKI and AAA Server Integration

The table below lists the attribute-value (AV) pairs that are to be used when setting up PKI integration with a AAA server. (Note the values shown in the table are possible values.) The AV pairs must match the client configuration. If they do not match, the peer certificate is not authorized.

Users can sometimes have AV pairs that are different from those of every other user. As a result, a unique username is required for each user. The all parameter (within the authorization username command) specifies that the entire subject name of the certificate will be used as the authorization username.

Table 2: AV Pairs That Must Match

<table>
<thead>
<tr>
<th>AV Pair</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>cisco-avpair=pki:cert-application=all</td>
<td>Valid values are &quot;all&quot; and &quot;none.&quot;</td>
</tr>
<tr>
<td>cisco-avpair=pki:cert-trustpoint=msca</td>
<td>The value is a Cisco IOS command-line interface (CLI) configuration trustpoint label.</td>
</tr>
<tr>
<td></td>
<td>The cert-trustpoint AV pair is normally optional. If it is specified, the Cisco IOS router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.</td>
</tr>
<tr>
<td></td>
<td>Note</td>
</tr>
<tr>
<td></td>
<td>The cert-serial AV pair is normally optional. If it is specified, the Cisco IOS router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.</td>
</tr>
<tr>
<td>cisco-avpair=pki:cert-serial=16318DB7000100001671</td>
<td>The value is a certificate serial number.</td>
</tr>
<tr>
<td>cisco-avpair=pki:cert-lifetime-end=1:00 jan 1, 2003</td>
<td>The cert-lifetime-end AV pair is available to artificially extend a certificate lifetime beyond the time period that is indicated in the certificate itself. If the cert-lifetime-end AV pair is used, the cert-trustpoint and cert-serial AV pairs must also be specified. The value must match the following form: hours:minutes month day, year.</td>
</tr>
<tr>
<td></td>
<td>Note</td>
</tr>
<tr>
<td></td>
<td>Only the first three characters of a month are used: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. If more than three characters are entered for the month, the remaining characters are ignored (for example Janxxxx).</td>
</tr>
</tbody>
</table>
CRLs or OCSP Server Choosing a Certificate Revocation Mechanism

After a certificate is validated as a properly signed certificate, a certificate revocation method is performed to ensure that the certificate has not been revoked by the issuing CA. Cisco IOS software supports two revocation mechanisms—certificate revocation lists (CRLs) and Online Certificate Status Protocol (OCSP). Cisco IOS software also supports AAA integration for certificate checking; however, additional authorization functionality is included. For more information on PKI and AAA certificate authorization and status check, see the PKI and AAA Server Integration for Certificate Status section.

The following sections explain how each revocation mechanism works:

What Is a CRL

A certificate revocation list (CRL) is a list of revoked certificates. The CRL is created and digitally signed by the CA that originally issued the certificates. The CRL contains dates for when each certificate was issued and when it expires.

CAs publish new CRLs periodically or when a certificate for which the CA is responsible has been revoked. By default, a new CRL is downloaded after the currently cached CRL expires. An administrator may also configure the duration for which CRLs are cached in router memory or disable CRL caching completely. The CRL caching configuration applies to all CRLs associated with a trustpoint.

When the CRL expires, the router deletes it from its cache. A new CRL is downloaded when a certificate is presented for verification; however, if a newer version of the CRL that lists the certificate under examination is on the server but the router is still using the CRL in its cache, the router does not know that the certificate has been revoked. The certificate passes the revocation check even though it should have been denied.

When a CA issues a certificate, the CA can include in the certificate the CRL distribution point (CDP) for that certificate. Cisco IOS client devices use CDPs to locate and load the correct CRL. The Cisco IOS client supports multiple CDPs, but the Cisco IOS CA currently supports only one CDP; however, third-party vendor CAs may support multiple CDPs or different CDPs per certificate. If a CDP is not specified in the certificate, the client device uses the default Simple Certificate Enrollment Protocol (SCEP) method to retrieve the CRL. (The CDP location can be specified through the cdp-url command.)

When implementing CRLs, you should consider the following design considerations:

- CRL lifetimes and the security association (SA) and Internet Key Exchange (IKE) lifetimes.
- The CRL lifetime determines the length of time between CA-issued updates to the CRL. The default CRL lifetime value, which is 168 hours [1 week], can be changed through the lifetime crl command.
- The method of the CDP determines how the CRL is retrieved; some possible choices include HTTP, Lightweight Directory Access Protocol (LDAP), SCEP, or TFTP. HTTP, TFTP, and LDAP are the most commonly used methods. Although Cisco IOS software defaults to SCEP, an HTTP CDP is recommended for large installations using CRLs because HTTP can be made highly scalable.
- The location of the CDP determines from where the CRL is retrieved; for example, you can specify the server and file path from which to retrieve the CRL.

Querying All CDPs During Revocation Check

When a CDP server does not respond to a request, the Cisco IOS software reports an error, which may result in the peer’s certificate being rejected. To prevent a possible certificate rejection and if there are multiple
CDPs in a certificate, the Cisco IOS software will attempt to use the CDPs in the order in which they appear in the certificate. The router will attempt to retrieve a CRL using each CDP URL or directory specification. If an error occurs using a CDP, an attempt will be made using the next CDP.

**Note**

Prior to Cisco IOS Release 12.3(7)T, the Cisco IOS software makes only one attempt to retrieve the CRL, even when the certificate contains more than one CDP.

**Tip**

Although the Cisco IOS software will make every attempt to obtain the CRL from one of the indicated CDPs, it is recommended that you use an HTTP CDP server with high-speed redundant HTTP servers to avoid application timeouts because of slow CDP responses.

**What Is OCSP**

OCSP is an online mechanism that is used to determine certificate validity and provides the following flexibility as a revocation mechanism:

- OCSP can provide real-time certificate status checking.
- OCSP allows the network administrator to specify a central OCSP server, which can service all devices within a network.
- OCSP also allows the network administrator the flexibility to specify multiple OCSP servers, either per client certificate or per group of client certificates.
- OCSP server validation is usually based on the root CA certificate or a valid subordinate CA certificate, but may also be configured so that external CA certificates or self-signed certificates may be used. Using external CA certificates or self-signed certificates allows the OCSP servers certificate to be issued and validated from an alternative PKI hierarchy.

A network administrator can configure an OCSP server to collect and update CRLs from different CA servers. The devices within the network can rely on the OCSP server to check the certificate status without retrieving and caching each CRL for every peer. When peers have to check the revocation status of a certificate, they send a query to the OCSP server that includes the serial number of the certificate in question and an optional unique identifier for the OCSP request, or a nonce. The OCSP server holds a copy of the CRL to determine if the CA has listed the certificate as being revoked; the server then responds to the peer including the nonce. If the nonce in the response from the OCSP server does not match the original nonce sent by the peer, the response is considered invalid and certificate verification fails. The dialog between the OCSP server and the peer consumes less bandwidth than most CRL downloads.

If the OCSP server is using a CRL, CRL time limitations will be applicable; that is, a CRL that is still valid might be used by the OCSP server although a new CRL has been issued by the CRL containing additional certificate revocation information. Because fewer devices are downloading the CRL information on a regular basis, you can decrease the CRL lifetime value or configure the OCSP server not to cache the CRL. For more information, check your OCSP server documentation.

**When to Use an OCSP Server**

OCSP may be more appropriate than CRLs if your PKI has any of the following characteristics:
• Real-time certificate revocation status is necessary. CRLs are updated only periodically and the latest CRL may not always be cached by the client device. For example, if a client does not yet have the latest CRL cached and a newly revoked certificate is being checked, that revoked certificate will successfully pass the revocation check.

• There are a large number of revoked certificates or multiple CRLs. Caching a large CRL consumes large portions of Cisco IOS memory and may reduce resources available to other processes.

• CRLs expire frequently, causing the CDP to handle a larger load of CRLs.

As of Cisco IOS Release 12.4(9)T or later, an administrator may configure CRL caching, either by disabling CRL caching completely or setting a maximum lifetime for a cached CRL per trustpoint.

When to Use Certificate-Based ACLs for Authorization or Revocation

Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action.

Because certificate-based ACLs are configured on the device, they do not scale well for large numbers of ACLs; however, certificate-based ACLs do provide very granular control of specific device behavior. Certificate-based ACLs are also leveraged by additional features to help determine when PKI components such as revocation, authorization, or a trustpoint should be used. They provide a general mechanism allowing users to select a specific certificate or a group of certificates that are being validated for either authorization or additional processing.

Certificate-based ACLs specify one or more fields within the certificate and an acceptable value for each specified field. You can specify which fields within a certificate should be checked and which values those fields may or may not have.

There are six logical tests for comparing the field with the value--equal, not equal, contains, does not contain, less than, and greater than or equal. If more than one field is specified within a single certificate-based ACL, the tests of all of the fields within the ACL must succeed to match the ACL. The same field may be specified multiple times within the same ACL. More than one ACL may be specified, and ACL will be processed in turn until a match is found or all of the ACLs have been processed.

Ignore Revocation Checks Using a Certificate-Based ACL

Certificate-based ACLs can be configured to instruct your router to ignore the revocation check and expired certificates of a valid peer. Thus, a certificate that meets the specified criteria can be accepted regardless of the validity period of the certificate, or if the certificate meets the specified criteria, revocation checking does not have to be performed. You can also use a certificate-based ACL to ignore the revocation check when the communication with a AAA server is protected with a certificate.

Ignoring Revocation Lists

To allow a trustpoint to enforce CRLs except for specific certificates, enter the match certificate command with the skip revocation-check keyword. This type of enforcement is most useful in a hub-and-spoke configuration in which you also want to allow direct spoke-to-spoke connections. In pure hub-and-spoke configurations, all spokes connect only to the hub, so CRL checking is necessary only on the hub. For one spoke to communicate directly with another spoke, the match certificate command with the skip
revocation-check keyword can be used for neighboring peer certificates instead of requiring a CRL on each spoke.

**Ignoring Expired Certificates**

To configure your router to ignore expired certificates, enter the `match certificate` command with the `allow expired-certificate` keyword. This command has the following purposes:

- If the certificate of a peer has expired, this command may be used to "allow" the expired certificate until the peer can obtain a new certificate.

- If your router clock has not yet been set to the correct time, the certificate of a peer will appear to be not yet valid until the clock is set. This command may be used to allow the certificate of the peer even though your router clock is not set.

**Note**

If Network Time Protocol (NTP) is available only via the IPSec connection (usually via the hub in a hub-and-spoke configuration), the router clock can never be set. The tunnel to the hub cannot be "brought up" because the certificate of the hub is not yet valid.

• "Expired" is a generic term for a certificate that is expired or that is not yet valid. The certificate has a start and end time. An expired certificate, for purposes of the ACL, is one for which the current time of the router is outside the start and end times specified in the certificate.

**Skipping the AAA Check of the Certificate**

If the communication with an AAA server is protected with a certificate, and you want to skip the AAA check of the certificate, use the `match certificate` command with the `skip authorization-check` keyword. For example, if a virtual private network (VPN) tunnel is configured so that all AAA traffic goes over that tunnel, and the tunnel is protected with a certificate, you can use the `match certificate` command with the `skip authorization-check` keyword to skip the certificate check so that the tunnel can be established.

The `match certificate` command and the `skip authorization-check` keyword should be configured after PKI integration with an AAA server is configured.

**Note**

If the AAA server is available only via an IPSec connection, the AAA server cannot be contacted until after the IPSec connection is established. The IPSec connection cannot be "brought up" because the certificate of the AAA server is not yet valid.

**PKI Certificate Chain Validation**

A certificate chain establishes a sequence of trusted certificates --from a peer certificate to the root CA certificate. Within a PKI hierarchy, all enrolled peers can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA. Each CA corresponds to a trustpoint.

When a certificate chain is received from a peer, the default processing of a certificate chain path continues until the first trusted certificate, or trustpoint, is reached. In Cisco IOS Release 12.4(6)T and later releases,
an administrator may configure the level to which a certificate chain is processed on all certificates including subordinate CA certificates.

Configuring the level to which a certificate chain is processed allows for the reauthentication of trusted certificates, the extension of a trusted certificate chain, and the completion of a certificate chain that contains a gap.

**Reauthentication of Trusted Certificates**

The default behavior is for the router to remove any trusted certificates from the certificate chain sent by the peer before the chain is validated. An administrator may configure certificate chain path processing so that the router does not remove CA certificates that are already trusted before chain validation, so that all certificates in the chain are re-authenticated for the current session.

**Extending the Trusted Certificate Chain**

The default behavior is for the router to use its trusted certificates to extend the certificate chain if there are any missing certificates in the certificate chain sent by the peer. The router will validate only certificates in the chain sent by the peer. An administrator may configure certificate chain path processing so that the certificates in the peer’s certificate chain and the router’s trusted certificates are validated to a specified point.

**Completing Gaps in a Certificate Chain**

An administrator may configure certificate chain processing so that if there is a gap in the configured Cisco IOS trustpoint hierarchy, certificates sent by the peer can be used to complete the set of certificates to be validated.

---

**Note**

If the trustpoint is configured to require parent validation and the peer does not provide the full certificate chain, the gap cannot be completed and the certificate chain is rejected and invalid.

**Note**

It is a configuration error if the trustpoint is configured to require parent validation and there is no parent trustpoint configured. The resulting certificate chain gap cannot be completed and the subordinate CA certificate cannot be validated. The certificate chain is invalid.

---

**High-Availability Support**

High-availability support for the certificate server is provided by:

- Synchronizing revoke commands with the standby certificate server
- Sending serial-number commands when new certificates are issued

The means that the standby certificate server is ready to issue certificates and certificate revocation lists (CRLs) if it becomes active.

Further high-availability support is provided by the following synchronizations with the standby:

- Certificate-server configuration
- Pending requests
• Grant and reject commands
• For box-to-box high availability, which does not support configuration synchronization, a basic configuration synchronization mechanism is layered over a redundancy facility.
• Trustpoint configuration synchronization support.

How to Configure Authorization and Revocation of Certificates for Your PKI

Configuring PKI Integration with a AAA Server

Perform this task to generate a AAA username from the certificate presented by the peer and specify which fields within a certificate should be used to build the AAA database username.

The following restrictions should be considered when using the **all** keyword as the subject name for the **authorization username** command:

- Some AAA servers limit the length of the username (for example, to 64 characters). As a result, the entire certificate subject name cannot be longer than the limitation of the server.
- Some AAA servers limit the available character set that may be used for the username (for example, a space [ ] and an equal sign [=] may not be acceptable). You cannot use the **all** keyword for a AAA server having such a character-set limitation.
- The **subject-name** command in the trustpoint configuration may not always be the final AAA subject name. If the fully qualified domain name (FQDN), serial number, or IP address of the router are included in a certificate request, the subject name field of the issued certificate will also have these components. To turn off the components, use the **fqdn**, **serial-number**, and **ip-address** commands with the **none** keyword.
- CA servers sometimes change the requested subject name field when they issue a certificate. For example, CA servers of some vendors switch the relative distinguished names (RDNs) in the requested subject names to the following order: CN, OU, O, L, ST, and C. However, another CA server might append the configured LDAP directory root (for example, O=cisco.com) to the end of the requested subject name.
- Depending on the tools you choose for displaying a certificate, the printed order of the RDNs in the subject name could be different. Cisco IOS software always displays the least significant RDN first, but other software, such as Open Source Secure Socket Layer (OpenSSL), does the opposite. Therefore, if you are configuring a AAA server with a full distinguished name (DN) (subject name) as the corresponding username, ensure that the Cisco IOS software style (that is, with the least significant RDN first) is used.

```bash
radius-server host hostname [key string]
```
SUMMARY STEPS

1. enable
2. configure terminal
3. aaa new-model
4. aaa authorization network listname [method]
5. crypto pki trustpoint name
6. enrollment [mode] [retry period minutes] [retry count number] url url [pem]
7. revocation-check method
8. exit
9. authorization username subjectname subjectname
10. authorization list listname
11. tacacs-server host hostname [key string]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Router&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> aaa new-model</td>
<td>Enables the AAA access control model.</td>
</tr>
<tr>
<td>Example: Router(config)# aaa new-model</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> aaa authorization network listname [method]</td>
<td>Sets the parameters that restrict user access to a network.</td>
</tr>
<tr>
<td>Example: Router (config)# aaa authorization network maxaaa group tacacs+</td>
<td>• method --Can be group radius, group tacacs+, or group group-name.</td>
</tr>
<tr>
<td><strong>Step 5</strong> crypto pki trustpoint name</td>
<td>Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td>Example: Route (config)# crypto pki trustpoint msc</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 6    | `enrollment [mode] [retry period minutes] [retry count number] url url [pem]`  | Specifies the following enrollment parameters of the CA:  
  - (Optional) The `mode` keyword specifies the registration authority (RA) mode, if your CA system provides an RA. By default, RA mode is disabled.  
  - (Optional) The `retry period` keyword and `minutes` argument specifies the period, in minutes, in which the router waits before sending the CA another certificate request. Valid values are from 1 to 60. The default is 1.  
  - (Optional) The `retry count` keyword and `number` argument specifies the number of times a router will resend a certificate request when it does not receive a response from the previous request. Valid values are from 1 to 100. The default is 10.  
  - The `url` argument is the URL of the CA to which your router should send certificate requests.  
  - (Optional) The `pem` keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.  

**Example:**  
```  
Router (ca-trustpoint)# enrollment url http://caserver.myexample.com  
Router (ca-trustpoint)# enrollment url http://[2001:DB8:1:1::1]:80  
```

| 7    | `revocation-check method`  | (Optional) Checks the revocation status of a certificate.  

**Example:**  
```  
Router (ca-trustpoint)# revocation-check crl  
```

| 8    | `exit`  | Exits ca-trustpoint configuration mode and returns to global configuration mode.  

**Example:**  
```  
Router (ca-trustpoint)# exit  
```

| 9    | `authorization username subjectname subjectname`  | Sets parameters for the different certificate fields that are used to build the AAA username.  
  
  The `subjectname` argument can be any of the following:  
  - `all` -- Entire distinguished name (subject name) of the certificate.  
  - `commonname` -- Certification common name.  
  - `country` -- Certificate country.  
  - `email` -- Certificate e-mail.  

**Example:**  
```  
Router (config)# authorization username subjectname subjectname serialnumber  
```
### Configuring Authorization and Revocation of Certificates in a PKI

**Command or Action**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ipaddress --Certificate IP address.</td>
<td></td>
</tr>
<tr>
<td>• locality --Certificate locality.</td>
<td></td>
</tr>
<tr>
<td>• organization --Certificate organization.</td>
<td></td>
</tr>
<tr>
<td>• organizationalunit --Certificate organizational unit.</td>
<td></td>
</tr>
<tr>
<td>• postalcode --Certificate postal code.</td>
<td></td>
</tr>
<tr>
<td>• serialnumber --Certificate serial number.</td>
<td></td>
</tr>
<tr>
<td>• state --Certificate state field.</td>
<td></td>
</tr>
<tr>
<td>• streetaddress --Certificate street address.</td>
<td></td>
</tr>
<tr>
<td>• title --Certificate title.</td>
<td></td>
</tr>
<tr>
<td>• unstructuredname --Certificate unstructured name.</td>
<td></td>
</tr>
</tbody>
</table>

**Step 10**

**authorization list listname**

Example:

```
Router (config)# authorization list maxaaa
```

Specifies the AAA authorization list.

**Step 11**

**tacacs-server host hostname [key string]**

Example:

```
Router(config)# tacacs-server host 192.0.2.2 key a_secret_key
```

Specifies a TACACS+ host.

or

```
Router(config)# tacacs-server host 192.0.2.1 key another_secret_key
```

Specifies a RADIUS host.

**Troubleshooting Tips**

To display debug messages for the trace of interaction (message type) between the CA and the router, use the `debug crypto pki transactions` command. (See the sample output, which shows a successful PKI integration with AAA server exchange and a failed PKI integration with AAA server exchange.)
Successful Exchange

Router# debug crypto pki transactions
Apr 22 23:15:03.695: CRYPTO_PKI: Found a issuer match
Apr 22 23:15:03.955: CRYPTO_PKI: cert revocation status unknown.
Apr 22 23:15:03.955: CRYPTO_PKI: Certificate validated without revocation check

Each line that shows "CRYPTO_PKI_AAA" indicates the state of the AAA authorization checks. Each of the AAA AV pairs is indicated, and then the results of the authorization check are shown.

Failed Exchange

Router# debug crypto pki transactions
Apr 22 23:15:04.019: CRYPTO_PKI_AAA: checking AAA authorization (ipseccca_script_aal1ist, PKI_AAA-L, <all>)
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: authorization passed

In the above failed exchange, the certificate has expired.

Configuring a Revocation Mechanism for PKI Certificate Status Checking

Perform this task to set up a CRL as the certificate revocation mechanism--CRLs or OCSP--that is used to check the status of certificates in a PKI.

The revocation-check Command

Use the `revocation-check` command to specify at least one method (OCSP, CRL, or skip the revocation check) that is to be used to ensure that the certificate of a peer has not been revoked. For multiple methods, the order in which the methods are applied is determined by the order specified via this command.

If your router does not have the applicable CRL and is unable to obtain one or if the OCSP server returns an error, your router will reject the peer’s certificate--unless you include the `none` keyword in your configuration. If the `none` keyword is configured, a revocation check will not be performed and the certificate will always be accepted.

Nonces and Peer Communications with OCSP Servers

When using OCSP, nonces, unique identifiers for OCSP requests, are sent by default during peer communications with your OCSP server. The use of nonces offers a more secure and reliable communication channel between the peer and OCSP server.

If your OCSP server does not support nonces, you may disable the sending of nonces. For more information, check your OCSP server documentation.
Before You Begin

- Before issuing any client certificates, the appropriate settings on the server (such as setting the CDP) should be configured.

- When configuring an OCSP server to return the revocation status for a CA server, the OCSP server must be configured with an OCSP response signing certificate that is issued by that CA server. Ensure that the signing certificate is in the correct format, or the router will not accept the OCSP response. See your OCSP manual for additional information.

Note

- OCSP transports messages over HTTP, so there may be a time delay when you access the OCSP server.

- If the OCSP server depends on normal CRL processing to check revocation status, the same time delay that affects CRLs will also apply to OCSP.

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. oesp url url
5. revocation-check method1 [method2 method3]
6. oesp disable-nonce
7. exit
8. exit
9. show crypto pki certificates
10. show crypto pki trustpoints [status | label [status]]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>crypto pki trustpoint name</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# crypto pki trustpoint hazel</td>
</tr>
<tr>
<td></td>
<td>Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>ocsp url url</td>
</tr>
</tbody>
</table>
| Example:         | Router(ca-trustpoint)# ocsp url http://ocsp-server | (Optional) Specifies that a nonce, or an OCSP request unique identifier, will not be sent during peer communications with the OCSP server.
<p>|                  | - or - |
|                  | Router(ca-trustpoint)# ocsp url <a href="http://10.10.10.1:80">http://10.10.10.1:80</a> | Returns to privileged EXEC mode. |
|                  | - or - |
|                  | Router(ca-trustpoint)# ocsp url http://[2001DB8:1:1::2]:80 | Returns to global configuration mode. |
| <strong>Step 5</strong>       | revocation-check method1 [method2 method3] |
| Example:         | Router(ca-trustpoint)# revocation-check ocsp none | Checks the revocation status of a certificate. |
|                  | • crl --Certificate checking is performed by a CRL. This is the default option. |
|                  | • none --Certificate checking is ignored. |
|                  | • ocsp --Certificate checking is performed by an OCSP server. |
|                  | If a second and third method are specified, each method will be used only if the previous method returns an error, such as a server being down. |
| <strong>Step 6</strong>       | ocsp disable-nonce |
| Example:         | Router(ca-trustpoint)# ocsp disable-nonce |
| <strong>Step 7</strong>       | exit |
| Example:         | Router(ca-trustpoint)# exit |
| <strong>Step 8</strong>       | exit |
| Example:         | Router(config)# exit |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong> show crypto pki certificates</td>
<td>(Optional) Displays information about your certificates.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# show crypto pki certificates</td>
</tr>
<tr>
<td><strong>Step 10</strong> show crypto pki trustpoints [status</td>
<td>label [status]]</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# show crypto pki trustpoints</td>
</tr>
</tbody>
</table>

### Configuring Certificate Authorization and Revocation Settings

Perform this task to specify a certificate-based ACL, to ignore revocation checks or expired certificates, to manually override the default CDP location, to manually override the OCSP server setting, to configure CRL caching, or to set session acceptance or rejection based on a certificate serial number, as appropriate.

#### Configuring Certificate-Based ACLs to Ignore Revocation Checks

To configure your router to use certificate-based ACLs to ignore revocation checks and expired certificates, perform the following steps:

- Identify an existing trustpoint or create a new trustpoint to be used when verifying the certificate of the peer. Authenticate the trustpoint if it has not already been authenticated. The router may enroll with this trustpoint if you want. Do not set optional CRLs for the trustpoint if you plan to use the match certificate command and skip revocation-check keyword.

- Determine the unique characteristics of the certificates that should not have their CRL checked and of the expired certificates that should be allowed.

- Define a certificate map to match the characteristics identified in the prior step.

- You can add the match certificate command and skip revocation-check keyword and the match certificate command and allow expired-certificate keyword to the trustpoint that was created or identified in the first step.

**Note**

Certificate maps are checked even if the peer’s public key is cached. For example, when the public key is cached by the peer, and a certificate map is added to the trustpoint to ban a certificate, the certificate map is effective. This prevents a client with the banned certificate, which was once connected in the past, from reconnecting.
Manually Overriding CDPs in a Certificate

Users can override the CDPs in a certificate with a manually configured CDP. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate’s CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP.

Manually Overriding the OCSP Server Setting in a Certificate

Administrators can override the OCSP server setting specified in the Authority Information Access (AIA) field of the client certificate or set by the issuing the `ocsp_url` command. One or more OCSP servers may be manually specified, either per client certificate or per group of client certificates by the `match certificate override ocsp` command. The `match certificate override ocsp` command overrides the client certificate AIA field or the `ocsp_url` command setting if a client certificate is successfully matched to a certificate map during the revocation check.

**Note**

Only one OCSP server can be specified per client certificate.

Configuring CRL Cache Control

By default, a new CRL will be downloaded after the currently cached CRL expires. Administrators can either configure the maximum amount of time in minutes a CRL remains in the cache by issuing the `crl_cache delete-after` command or disable CRL caching by issuing the `crl_cache none` command. Only the `crl-cache delete-after` command or the `crl-cache none` command may be specified. If both commands are entered for a trustpoint, the last command executed will take effect and a message will be displayed.

Neither the `crl-cache none` command nor the `crl-cache delete-after` command affects the currently cached CRL. If you configure the `crl-cache none` command, all CRLs downloaded after this command is issued will not be cached. If you configure the `crl-cache delete-after` command, the configured lifetime will only affect CRLs downloaded after this command is issued.

This functionality is useful when a CA issues CRLs with no expiration date or with expiration dates days or weeks ahead.

Configuring Certificate Serial Number Session Control

A certificate serial number can be specified to allow a certificate validation request to be accepted or rejected by the trustpoint for a session. A session may be rejected, depending on certificate serial number session control, even if a certificate is still valid. Certificate serial number session control may be configured by using either a certificate map with the `serial-number` field or an AAA attribute, with the `cert-serial-not` command.

Using certificate maps for session control allows an administrator to specify a single certificate serial number. Using the AAA attribute allows an administrator to specify one or more certificate serial numbers for session control.

**Before You Begin**

- The trustpoint should be defined and authenticated before attaching certificate maps to the trustpoint.
The certificate map must be configured before the CDP override feature can be enabled or the serial-number command is issued.

The PKI and AAA server integration must be successfully completed to use AAA attributes as described in "PKI and AAA Server Integration for Certificate Status."

**SUMMARY STEPS**

1. enable
2. configure terminal
3. crypto pki certificate map label sequence-number
4. field-name match-criteria match-value
5. exit
6. crypto pki trustpoint name
7. Do one of the following:
   - crl-cache none
   - crl-cache delete-after time
8. match certificate certificate-map-label [allow expired-certificate | skip revocation-check | skip authorization-check]
9. match certificate certificate-map-label override cdp {url | directory} string
10. match certificate certificate-map-label override ocsp [trustpoint trustpoint-label] sequence-number url ocsp-url
11. exit
12. aaa new-model
13. aaa attribute list list-name
14. attribute type {name} {value}
15. exit
16. exit
17. show crypto pki certificates

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>enable</td>
<td></td>
</tr>
<tr>
<td>Example: Router&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

- Enter your password if prompted.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto pki certificate map label sequence-number</td>
<td>Defines values in a certificate that should be matched or not matched and enters ca-certificate-map configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# crypto pki certificate map Group 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> field-name match-criteria match-value</td>
<td>Specifies one or more certificate fields together with their matching criteria and the value to match.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(ca-certificate-map)# subject-name co MyExample</td>
<td></td>
</tr>
</tbody>
</table>

The **field-name** is one of the following case-insensitive name strings or a date:

- alt-subject-name
- expires-on
- issuer-name
- name
- serial-number
- subject-name
- unstructured-subject-name
- valid-start

**Note** Date field format is dd mm yyyy hh:mm:ss or mmm dd yyyy hh:mm:ss.

The **match-criteria** is one of the following logical operators:

- co --contains (valid only for name fields and serial number field)
- eq --equal (valid for name, serial number, and date fields)
- ge --greater than or equal (valid only for date fields)
- lt --less than (valid only for date fields)
- nc --does not contain (valid only for name fields and serial number field)
- ne --not equal (valid for name, serial number, and date fields)

The **match-value** is the name or date to test with the logical operator assigned by match-criteria.

**Note** Use this command only when setting up a certificate-based ACL--not when setting up a certificate-based ACL to ignore revocation checks or expired certificates.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example:</td>
<td>Returns to global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Router(ca-certificate-map)# exit</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>crypto pki trustpoint name</td>
</tr>
<tr>
<td>Example:</td>
<td>Declares the trustpoint, given name and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Router(config)# crypto pki trustpoint Access2</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Do one of the following:</td>
</tr>
<tr>
<td></td>
<td>• crl-cache none</td>
</tr>
<tr>
<td></td>
<td>• crl-cache delete-after time</td>
</tr>
<tr>
<td>Example:</td>
<td>(Optional) Disables CRL caching completely for all CRLs associated with the trustpoint.</td>
</tr>
<tr>
<td></td>
<td>The crl-cache none command does not affect any currently cached CRLs. All CRLs downloaded after this command is configured will not be cached.</td>
</tr>
<tr>
<td></td>
<td>(Optional) Specifies the maximum time CRLs will remain in the cache for all CRLs associated with the trustpoint.</td>
</tr>
<tr>
<td></td>
<td>• time -- The amount of time in minutes before the CRL is deleted.</td>
</tr>
<tr>
<td></td>
<td>The crl-cache delete-after command does not affect any currently cached CRLs. The configured lifetime will only affect CRLs downloaded after this command is configured.</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>match certificate certificate-map-label allow expired-certificate skip revocation-check skip authorization-check</td>
</tr>
<tr>
<td>Example:</td>
<td>(Optional) Associates the certificate-based ACL (that was defined via the crypto pki certificate map command) to a trustpoint.</td>
</tr>
<tr>
<td></td>
<td>• certificate-map-label -- Must match the label argument specified via the crypto pki certificate map command.</td>
</tr>
<tr>
<td></td>
<td>• allow expired-certificate -- Ignores expired certificates.</td>
</tr>
<tr>
<td></td>
<td>• skip revocation-check -- Allows a trustpoint to enforce CRLs except for specific certificates.</td>
</tr>
<tr>
<td></td>
<td>• skip authorization-check -- Skips the AAA check of a certificate when PKI integration with an AAA server is configured.</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>match certificate certificate-map-label override cdp {url</td>
</tr>
<tr>
<td>Example:</td>
<td>(Optional) Manually overrides the existing CDP entries for a certificate with a URL or directory specification.</td>
</tr>
<tr>
<td></td>
<td>• certificate-map-label -- A user-specified label that must match the label argument specified in a previously defined crypto pki certificate map command.</td>
</tr>
<tr>
<td></td>
<td>• url -- Specifies that the certificate’s CDPs will be overridden with an HTTP or LDAP URL.</td>
</tr>
</tbody>
</table>

**Public Key Infrastructure Configuration Guide, Cisco IOS Release 15S**
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **match certificate**  *certificate-map-label*  
**override**  *ocsp*  
**trustpoint**  *trustpoint-label*  
**sequence-number**  *url*  
**ocsp-url** | *(Optional) Specifies an OCSP server, either per client certificate or per group of client certificates, and may be issued more than once to specify additional OCSP servers and client certificate settings including alternative PKI hierarchies.*

**Note**

Some applications may time out before all CDPs have been tried and will report an error message. The error message will not affect the router, and the Cisco IOS software will continue attempting to retrieve a CRL until all CDPs have been tried.

**Example:**

```
Router(ca-trustpoint)# match certificate mycertmapname override ocsp trustpoint mytp 15 url http://192.0.2.2
```

**directory** -- Specifies that the certificate’s CDPs will be overridden with an LDAP directory specification.

**string** -- The URL or directory specification.

**Step 10**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>exit</strong></td>
<td>Returns to global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(ca-trustpoint)# exit
```

**Step 12**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>aaa new-model</strong></td>
<td><em>(Optional) Enables the AAA access control model.</em></td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config)# aaa new-model
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 13** | **aaa attribute list** *list-name*  
**Example:**  
Router(config)# aaa attribute list crl  
(Optional) Defines an AAA attribute list locally on a router and enters config-attr-list configuration mode. |
| **Step 14** | **attribute type** `{name}` `{value}`  
**Example:**  
Router(config-attr-list)# attribute type cert-serial-not 6C4A  
(Optional) Defines an AAA attribute type that is to be added to an AAA attribute list locally on a router.  
To configure certificate serial number session control, an administrator may specify a specific certificate in the `value` field to be accepted or rejected based on its serial number where `name` is set to `cert-serial-not`. If the serial number of the certificate matches the serial number specified by the attribute type setting, the certificate will be rejected.  
For a full list of available AAA attribute types, execute the `show aaa attributes` command. |
| **Step 15** | **exit**  
**Example:**  
Router(ca-trustpoint)# exit  
Example:  
Router(config-attr-list)# exit  
Returns to global configuration mode. |
| **Step 16** | **exit**  
**Example:**  
Router(config)# exit  
Returns to privileged EXEC mode. |
| **Step 17** | **show crypto pki certificates**  
**Example:**  
Router# show crypto pki certificates  
(Optional) Displays the components of the certificates installed on the router if the CA certificate has been authenticated. |

**Example**

The following is a sample certificate. The OCSP-related extensions are shown using exclamation points.

```
Certificate:  
  Data:  
    Version: v3  
    Serial Number:0x14  
    Signature Algorithm:SHAwithRSA - 1.2.840.113549.1.1.4  
    Issuer:CN=CA server,OU=PKI,O=Cisco Systems
```
Validity:
  Not Before: Thursday, August 8, 2002 4:38:05 PM PST
  Not After: Tuesday, August 7, 2003 4:38:05 PM PST
Subject: CN=OCSP server, OU=PKI, O=Cisco Systems
Subject Public Key Info:
  Algorithm: RSA - 1.2.840.113549.1.1.1
  Public Key:
    Exponent: 65537
    Public Key Modulus: (2048 bits):
      <snip>
Extensions:
  Identifier: Subject Key Identifier = 2.5.29.14
    Critical: no
    Key Identifier:
      <snip>
  Identifier: Authority Key Identifier = 2.5.29.35
    Critical: no
    Key Identifier:
      <snip>
  Identifier: OCSP NoCheck = 1.3.6.1.5.5.7.48.1.5
    Critical: no
  Identifier: Extended Key Usage = 2.5.29.37
    Critical: no
    Extended Key Usage:
      OCSPSigning
  Identifier: CRL Distribution Points = 2.5.29.31
    Critical: no
    Number of Points: 1
    Point 0
      Distribution Point:
        [URIName: ldap://CA-server/CN=CA server, OU=PKI, O=Cisco Systems]
Signature:
  Algorithm: SHAwithRSA - 1.2.840.113549.1.1.4
  Signature:
    <snip>

The following example shows an excerpt of the running configuration output when adding a **match certificate override ocsp** command to the beginning of an existing sequence:

```plaintext
match certificate map3 override ocsp 5 url http://192.0.2.3/
show running-configuration
```

The following example shows an excerpt of the running configuration output when an existing **match certificate override ocsp** command is replaced and a trustpoint is specified to use an alternative PKI hierarchy:

```plaintext
match certificate map4 override ocsp trustpoint tp4 10 url http://192.0.2.4/newvalue
show running-configuration
```

**Troubleshooting Tips**

If you ignored revocation check or expired certificates, you should carefully check your configuration. Verify that the certificate map properly matches either the certificate or certificates that should be allowed or the
AAA checks that should be skipped. In a controlled environment, try modifying the certificate map and determine what is not working as expected.

**Configuring Certificate Chain Validation**

Perform this task to configure the processing level for the certificate chain path of your peer certificates.

**Before You Begin**

- The device must be enrolled in your PKI hierarchy.
- The appropriate key pair must be associated with the certificate.

**Note**

- A trustpoint associated with the root CA cannot be configured to be validated to the next level.

The `chain-validation` command is configured with the `continue` keyword for the trustpoint associated with the root CA, an error message will be displayed and the chain validation will revert to the default `chain-validation` command setting.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `crypto pki trustpoint name`
4. `chain-validation [{stop | continue} [parent-trustpoint]]`
5. `exit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Router&gt; enable</code></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Router# configure terminal</code></td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td>crypto pki trustpoint <em>name</em></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# crypto pki trustpoint ca-sub1</td>
</tr>
<tr>
<td>Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 4** | chain-validation [{stop | continue} [parent-trustpoint]] |
| **Example:** | Router(ca-trustpoint)# chain-validation continue ca-sub1 |
| Configures the level to which a certificate chain is processed on all certificates including subordinate CA certificates. |
| • Use the stop keyword to specify that the certificate is already trusted. This is the default setting. |
| • Use the continue keyword to specify that the that the subordinate CA certificate associated with the trustpoint must be validated. |
| • The parent-trustpoint argument specifies the name of the parent trustpoint the certificate must be validated against. |

| **Step 5** | exit |
| **Example:** | Router(ca-trustpoint)# exit |
| Returns to global configuration mode |

---

### Configuring Certificate Servers for High Availability

You can configure certificate servers to synchronize revoke commands and send serial-number commands when new certificates are issued, preparing the standby certificate server to issue certificates and CRLs if it becomes active.

#### Prerequisites

The following conditions must be met for high availability on certificate servers:

- IPsec-secured SCTP must be configured on both the active and the standby routers.
- For synchronization to work, the redundancy mode on the certificate servers must be set to ACTIVE/STANDBY after you configure SCTP.

This section contains the following subsections:

#### Setting Redundancy Mode on Certificate Servers to ACTIVE STANDBY

Perform this task on the active router to enable synchronization by setting the redundancy mode on the certificate servers to ACTIVE/STANDBY.
1 configure terminal
2 redundancy inter-device
3 scheme standby standby-group-name
4 exit
5 interface interface-name
6 ip address ip-address mask
7 no ip route-cache cef
8 no ip route-cache
9 standby ip ip-address
10 standby priority priority
11 standby name group-name
12 standby delay minimum [min-seconds] reload [reload-seconds]
13 Repeat Steps 1-12 on the standby router, r, configuring the interface with a different IP address from that of the active router (Step 6).
14 exit
15 exit
16 show crypto key mypubkey rsa

SUMMARY STEPS

1. configure terminal
2. redundancy inter-device
3. scheme standby standby-group-name
4. exit
5. interface interface-name
6. ip address ip-address mask
7. no ip route-cache cef
8. no ip route-cache
9. standby ip ip-address
10. standby priority priority
11. standby name group-name
12. standby delay minimum [min-seconds] reload [reload-seconds]
13. Repeat Steps 1-12 on the standby router, configuring the interface with a different IP address from that of the interface on the active router (Step 6).
14. exit
15. exit
16. show redundancy states
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>redundancy inter-device</td>
<td>Configures redundancy and enters interdevice configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Router(config)# redundancy inter-device</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>scheme standby <em>standby-group-name</em></td>
<td>Defines the redundancy scheme that is to be used.</td>
</tr>
</tbody>
</table>
|       | Example: Router(config-red-interdevice)# scheme standby SB | - The only supported scheme is “standby.”  
- *standby-group-name* --Must match the standby name specified in the *standby name* interface configuration command. Also, the standby name must be the same on both routers. |
<p>| <strong>Step 4</strong> | exit | Exits interdevice configuration mode and returns to global configuration mode. |
|       | Example: Router(config-red-interdevice)# exit | |
| <strong>Step 5</strong> | interface <em>interface-name</em> | Configures an interface type for the router and enters interface configuration mode. |
|       | Example: Router(config) # interface gigabitethernet0/1 | |
| <strong>Step 6</strong> | ip address <em>ip-address mask</em> | Sets the local IP address for the interface. |
|       | Example: Router(config-if) ip address 10.0.0.1 255.255.255.0 | |
| <strong>Step 7</strong> | no ip route-cache cef | Disables Cisco Express Forwarding operation on the interface. |
|       | Example: Router(config-if)# no ip route cache cef | |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 8</strong> no ip route-cache</td>
<td>Enables fast switching on the interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-if)# no ip route cache</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> standby ip ip-address</td>
<td>Activates the Hot Standby Router Protocol (HSRP), Note: Configure the same address on the active and the standby routers.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-if)# standby ip 10.0.0.3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> standby priority priority</td>
<td>Sets the HSRP priority to 50. The priority range is from 1 to 255, where 1 denotes the lowest priority and 255 the highest. The router in the HSRP group with the highest priority value becomes the active router.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-if)# standby priority 50</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> standby name group-name</td>
<td>Configures the name of the standby group. • The name specifies the HSRP group used. The HSRP group name must be unique on the router.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-if)# standby name SB</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> standby delay minimum [ min-seconds ] reload [ reload-seconds ]</td>
<td>Sets a delay for HSRP group initialization as follows: • The minimum delay after the interface comes up before initializing the HSRP groups is 30 seconds. • The delay after the router has reloaded is 60 seconds.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-if)# standby delay minimum 30 reload 60</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> Repeat Steps 1-12 on the standby router, configuring the interface with a different IP address from that of the interface on the active router (Step 6).</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 14</strong> exit</td>
<td>Returns to global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong> exit</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# exit</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

Command or Action | Purpose
---|---
Step 16 | show redundancy states
  
  Example:
  
  Router# show redundancy states

### Configuring SCTP on the Active and Standby Certificate Servers

Perform this task on the active router to configure SCTP on both the active and the standby certificate server.

#### SUMMARY STEPS

1. `configure terminal`
2. `ipc zone default`
3. `association association-ID`
4. `no shutdown`
5. `protocol sctp`
6. `local-port local-port-number`
7. `local-ip device-real-ip-address [device-real-ip-address2]`
8. `exit`
9. `remote-port remote-port-number`
10. `remote-ip peer-real-ip-address`
11. Repeat Steps 1 through 10 on the standby router, reversing the IP addresses of the local and remote peers specified in Steps 7 and 10.

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | `configure terminal`
  
  Example:
  
  Router# configure terminal |
  
  Enters global configuration mode. |
| **Step 2** | `ipc zone default`
  
  Example:
  
  Router(config)# ipc zone default |
  
  Configures the interdevice communication protocol, Inter-Process Communication (IPC), and enters IPC zone configuration mode.
  
  Use this command to initiate the communication link between the active router and the standby router. |
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>association 1</td>
<td>Configures an association between the two devices and enters IPC association configuration mode.</td>
</tr>
<tr>
<td></td>
<td>association-ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-ipczone)# association 1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>no shutdown</td>
<td>Ensures that the server association is in the default (enabled) state.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-ipczone-assoc)# no shutdown</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>protocol sctp</td>
<td>Configures SCTP as the transport protocol and enters SCTP protocol configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-ipczone-assoc)# protocol sctp</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>local-port 5000</td>
<td>Defines the local SCTP port number that is used to communicate with the redundant peer and enters IPC transport SCTP local configuration mode.</td>
</tr>
<tr>
<td></td>
<td>local-port-number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-ipc-protocol-sctp)# local-port 5000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>local-port-number</td>
<td>1 to 65535. The local port numbers should be the same as the remote port number on the peer router.</td>
</tr>
<tr>
<td>7</td>
<td>local-ip 10.0.0.1</td>
<td>Defines at least one local IP address that is used to communicate with the redundant peer.</td>
</tr>
<tr>
<td></td>
<td>device-real-ip-address [device-real-ip-address2]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-ipc-local-sctp)# local-ip 10.0.0.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The local IP addresses must match the remote IP addresses on the peer router. There can be either one or two IP addresses, which must be in global VPN routing and forwarding (VRF). A virtual IP address cannot be used.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>exit</td>
<td>Exits IPC transport - SCTP local configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-ipc-local-sctp)# exit</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>remote-port 5000</td>
<td>Defines the remote SCTP port number that is used to communicate with the redundant peer and enters IPC transport SCTP remote configuration mode.</td>
</tr>
<tr>
<td></td>
<td>remote-port-number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-ipc-protocol-sctp)# remote-port 5000</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>remote-port-number</td>
<td>remote-port-number -- There is not a default value. This argument must be configured for the remote port to enable interdevice redundancy. Valid port values: 1 to 65535. The remote port number should be the same as the local port number on the peer router.</td>
</tr>
</tbody>
</table>

### Step 10

**remote-ip peer-real-ip-address**

*Example:*

Router(config-ipc-remote-sctp)# remote-ip 10.0.0.2

- Defines a remote IP address of the redundant peer that is used to communicate with the local device.
- All remote IP addresses must refer to the same device.
- A virtual IP address cannot be used.

### Step 11

Repeat Steps 1 through 10 on the standby router, reversing the IP addresses of the local and remote peers specified in Steps 7 and 10.

The virtual IP address (10.0.0.3) will be the same on both routers.

---

### Synchronizing the Active and Standby Certificate Servers

Perform this task to synchronize the active and standby servers.

**SUMMARY STEPS**

1. configure terminal
2. crypto key generate rsa general-keys redundancy label key-labe modulus modulus-size
3. exit
4. show crypto key mypubkey rsa
5. configure terminal
6. ip http server
7. crypto pki server cs-label
8. redundancy
9. no shutdown

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

Router# configure terminal

**Step 2**

crypto key generate rsa general-keys redundancy label key-labe modulus modulus-size

Generates an RSA key pair named HA for the certificate server.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong> exit</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Step 4</strong> show crypto key mypubkey rsa</td>
<td>Verifies that redundancy is enabled.</td>
</tr>
<tr>
<td><strong>Step 5</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 6</strong> ip http server</td>
<td>Enables the HTTP server on your system.</td>
</tr>
<tr>
<td><strong>Step 7</strong> crypto pki server cs-label</td>
<td>Specifies the RSA key pair generated in Step 2 as the label for the certificate server and enters certificate server configuration mode.</td>
</tr>
<tr>
<td><strong>Step 8</strong> redundancy</td>
<td>Ensures that the server is synchronized to the standby server.</td>
</tr>
<tr>
<td><strong>Step 9</strong> no shutdown</td>
<td>Enables the certificate server.</td>
</tr>
</tbody>
</table>

**Note**

If the router interface with the SCTP traffic is not secure, you should ensure that the SCTP traffic between the high-availability devices is secured with IPsec.
**Configuration Examples for Setting Up Authorization and Revocation of Certificates**

**Configuring and Verifying PKI AAA Authorization Examples**

This section provides configuration examples of PKI AAA authorizations:

**Router Configuration Example**

The following `show running-config` command output shows the working configuration of a router that is set up to authorize VPN connections using the PKI Integration with AAA Server feature:

```
Router# show running-config
Building configuration...
!
version 12.3
!
hostname router7200
!
! aaa new-model
!
! aaa authentication login default group tacacs+
aaa authentication login no_tacacs enable
aaa authentication ppp default group tacacs+
aaa authentication exec ACSLab group tacacs+
aaa authorization network ACSLab group tacacs+
aaa authorization exec ACSLab start-stop group tacacs+
aaa accounting network default start-stop group ACSLab
aaa session-id common
!
ip domain name example.com
!
crypto pki trustpoint EM-CERT-SERV
enrollment url http://192.0.2.33:80
serial-number
crl optional
rsakeypair STOREVPN 2048
auto-enroll
authorization list ACSLab
!
crypto pki certificate chain EM-CERT-SERV
certificate 04
  30820214 3082017D A0030201 02020104 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403100C 4552542D 53455256 301E170D 30343031
  31393232 30233535 5A170030 35303131 38323230 32353535 3030312E 300E603
  35504513 07314437 45424434 301C0609 2A866886 F70D0109 02160F37 3230302D
  312E6672 696C2E63 656D6308 9F300D06 092A8668 6F670D00 01010500 0381BD00
  30819002 6218000D 3B38377A D925F391 2864DA14 9C2EA031 5A7203C4 92F5D6A8
  7D2357A6 BCC8596F A30A9B10 47435626 D59A8F2A 123195BB BE5AIE74 BIAA5AE0
  5CA162FF 8C3ACA4F B3EE9F27 80B31642 B618AE1B 40F2E3B4 F996BEFE 382C7283
  3792A369 236F8661 8748AA3F BC41F012 B859BD9C DB4F75EE 3CE32829 704D6B8F
  FD904043 0F555702 03010001 A3573055 30256030 551D1F04 1E301C30 1AA018A0
  1A814686 7474730A 2F2F3633 2E323437 2E323030 2E593330 0B060335 100F0400
  0310205A 301F0603 551D2304 18301680 1420FC48 CF081C56 F586DC06 0AF4D67
  341AE612 D1300006 092A8668 6F670D01 01040500 03818100 79E97018 FB955108
  12F42A55 26A384BC A36822FF F1D6187F DA5D6737 C0E241AC AAACC78D 3C74F59F
  08DEEF2 08B12A73 D79E0FA9 D62DC20D 8E2798CD 21CD5EC 3B2505A1 3897350C
  15A6050A 8A13F06D 51043D37 E56E45DF A65F437D 46836093 9689784D C45FDG1D
  ECF1F60C 1ABC8003 49FB11B1 DA0BED6C 463E1090 F34C59E4
```

Configuring Authorization and Revocation of Certificates in a PKI

Certificate ca 01
30820207 30820170 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
317311530 13060355 0403130C 454D2D43 4552542D 53455256 01E170D 30333132
31363231 34373432 5A170D30 36313231 35323134 3734325A 30173115 30130603
55040313 0C454D2D 43455254 2D534552 5630819F 300D0609 2A864886 F70D0101
01050003 81BD0030 81BD0281 810CC14D 833641CF D794F516 EAE5B6C1 75BC73C9
589223AB 9A70DC14 04F75E5E AAD55F8E E56989B6 F29691F7 D88986A1 2C726C69
54A29870 7E733F9F 3C019F91 F5A37CPF 3F6FD3D0 9E48C4C4 2A2E3459 0C2D0785
E9DE981E B73886F8 AAB9160C A8048607 D34B83C0 64D8D01C 161FC103 13C60500
22D6E747 76DCF133 7F15153F 32830203 010001A3 63306130 000F06355 1D310101
FF005050 030101FF 300E0603 551D0F01 01FF0404 03020186 03010001 000F06355
4107116B AD3AC990 CBE84063 5FB2A6B5 BD527026 52BE92ED 02F30AEE 1803F2AE
AA4086D5 6D6DEDC3 7F9C2C79 39635633 F298FB90 795BA8
quit
!

crypto isakmp policy 10
  encr aes
group 14
!
crypto ipsec transform-set ISC_TS_1 esp-aes esp-sha-hmac
! crypto ipsec profile ISC_IPSEC_PROFILE_2
  set security-association lifetime kilobytes 30000000
  set security-association lifetime seconds 14400
  set transform-set ISC_TS_1
!
controller ISA 1/1
!
interface Tunnel0
description MGRE Interface provisioned by ISC
bandwidth 10000
ip address 192.0.2.172 255.255.255.0
no ip redirects
ip mtu 1408
ip nhgrp multicast dynamic
ip nhgrp network-id 101
ip nhgrp holdtime 500
ip nhgrp server-only
no ip split-horizon eigrp 101
tunnel source FastEthernet2/1
tunnel mode gre multipoint
tunnel key 101
tunnel protection ipsec profile ISC_IPSEC_PROFILE_2
!
interface FastEthernet2/0
ip address 192.0.2.21 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet2/1
ip address 192.0.2.22 255.255.255.0
duplex auto
speed auto
!
tacacs-server host 192.0.2.55 single-connection
tacacs-server directed-request
tacacs-server key company lab
!
ntp master 1
!
end
Debug of a Successful PKI AAA Authorization Example

The following `show debugging` command output shows a successful authorization using the PKI Integration with AAA Server feature:

```
Router# show debugging
General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Cryptographic Subsystem:
  Crypto PKI Trans debugging is on
Router#
```
```
May 28 19:36:11.117: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:36:12.789: CRYPTO_PKI: Found a issuer match
May 28 19:36:12.813: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.example.com, <all>)
```
```
Router# show crypto isakmp sa
  dst  src  state  conn-id slot
  192.0.2.22 192.0.2.102  QM_IDLE  84  0
```

Debugs of a Failed PKI AAA Authorization Example

The following `show debugging` command output shows that the router is not authorized to connect using VPN. The messages are typical of those that you might see in such a situation.

In this example, the peer username was configured as not authorized, by moving the username to a Cisco Secure ACS group called VPN_Router_Disabled in Cisco Secure ACS. The router, router7200.example.com, has been configured to check with a Cisco Secure ACS AAA server prior to establishing a VPN connection to any peer.

```
Router# show debugging
General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Cryptographic Subsystem:
  Crypto PKI Trans debugging is on
Router#
```
```
May 28 19:36:18.681: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 101: Neighbor 192.0.2.171 (Tunnel0) is up: new adjacency
```
```
Router# show crypto isakmp sa
  dst  src  state  conn-id slot
  192.0.2.22 192.0.2.102  QM_IDLE  84  0
```
Configuring a Revocation Mechanism Examples

This section contains the following configuration examples that can be used when specifying a revocation mechanism for your PKI:
Configuring an OCSP Server Example

The following example shows how to configure the router to use the OCSP server that is specified in the AIA extension of the certificate:

Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check ocsp

Specifying a CRL and Then an OCSP Server Example

The following example shows how to configure the router to download the CRL from the CDP. If the CRL is unavailable, the OCSP server that is specified in the AIA extension of the certificate will be used. If both options fail, certificate verification will also fail.

Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check crl ocsp

Specifying an OCSP Server Example

The following example shows how to configure your router to use the OCSP server at the HTTP URL “http://myocspserver:81.” If the server is down, the revocation check will be ignored.

Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none

Disabling Nonces in Communications with the OCSP Server Example

The following example shows communications when a nonce, or a unique identifier for the OCSP request, is disabled for communications with the OCSP server:

Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
Router(ca-trustpoint)# ocsp disable-nonce

Configuring a Hub Router at a Central Site for Certificate Revocation Checks Example

The following example shows a hub router at a central site that is providing connectivity for several branch offices to the central site.

The branch offices are also able to communicate directly with each other using additional IPSec tunnels between the branch offices.

The CA publishes CRLs on an HTTP server at the central site. The central site checks CRLs for each peer when setting up an IPSec tunnel with that peer.

The example does not show the IPSec configuration—only the PKI-related configuration is shown.
Home Office Hub Configuration

crypto pki trustpoint VPN-GW
enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
serial-number none
fqdn none
ip-address none
subject-name o=Home Office Inc,cn=Central VPN Gateway
revocation-check crl

Central Site Hub Router

Router# show crypto ca certificate
Certificate
Status: Available
Certificate Serial Number: 2F62BE140000000000CA0
Certificate Usage: General Purpose
Issuer:
cn=Central Certificate Authority
o=Home Office Inc
Subject:
Name: Central VPN Gateway
cn=Central VPN Gateway
o=Home Office Inc
CRL Distribution Points:
http://ca.home-office.com/CertEnroll/home-office.crl
Validity Date:
start date: 00:43:26 GMT Sep 26 2003
date: 00:53:26 GMT Sep 26 2004
renew date: 00:00:00 GMT Jan 1 1970
Associated Trustpoints: VPN-GW

CA Certificate
Status: Available
Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
Certificate Usage: Signature
Issuer:
cn=Central Certificate Authority
o=Home Office Inc
Subject:
cn=Central Certificate Authority
o=Home Office Inc
CRL Distribution Points:
http://ca.home-office.com/CertEnroll/home-office.crl
Validity Date:
start date: 22:19:29 GMT Oct 31 2002
date: 22:27:27 GMT Oct 31 2017
Associated Trustpoints: VPN-GW

Trustpoint on the Branch Office Router

crypto pki trustpoint home-office
enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
serial-number none
fqdn none
ip-address none
subject-name o=Home Office Inc,cn=Branch 1
revocation-check crl

A certificate map is entered on the branch office router.

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
branch1(config)# crypto pki certificate map central-site 10
branch1(config)#
The output from the `show certificate` command on the central site hub router shows that the certificate was issued by the following:

```plaintext
cn=Central Certificate Authority
o=Home Office Inc
```

These two lines are combined into one line using a comma (,) to separate them, and the original lines are added as the first criteria for a match.

```plaintext
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home Office Inc
```

The above line wrapped but should be shown on one line with the line above it. The same combination is done for the subject name from the certificate on the central site router (note that the line that begins with “Name:” is not part of the subject name and must be ignored when creating the certificate map criteria). This is the subject name to be used in the certificate map.

```plaintext
cn=Central VPN Gateway
o=Home Office Inc
```

```plaintext
Router (ca-certificate-map)# subject-name eq cn=central vpn gateway, o=home office inc
```

Now the certificate map is added to the trustpoint that was configured earlier.

```plaintext
Router (ca-certificate-map)# crypto pki trustpoint home-office
Router (ca-trustpoint)# match certificate central-site skip revocation-check
Router (ca-trustpoint)# exit
Router (config)# exit
```

The configuration is checked (most of configuration is not shown).

```plaintext
Router# write term
```

```plaintext
crypto pki trustpoint home-office
enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
serial-number none
fqdn none
ip-address none
subject-name o=Home Office Inc,cn=Branch 1
revocation-check crl
match certificate central-site skip revocation-check
```

```plaintext
crypto pki certificate map central-site 10
issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
subject-name eq cn = central vpn gateway, o = home office inc
```

Note that the issuer-name and subject-name lines have been reformatted to make them consistent for later matching with the certificate of the peer.

If the branch office is checking the AAA, the trustpoint will have lines similar to the following:

```plaintext
crypto pki trustpoint home-office
auth list allow_list
auth user subj commonname
```

After the certificate map has been defined as was done above, the following command is added to the trustpoint to skip AAA checking for the central site hub.

```plaintext
match certificate central-site skip authorization-check
```

In both cases, the branch site router has to establish an IPSec tunnel to the central site to check CRLs or to contact the AAA server. However, without the `match certificate` command and `central-site skip authorization-check` (argument and keyword), the branch office cannot establish the tunnel until it has
checked the CRL or the AAA server. (The tunnel will not be established unless the **match certificate** command and **central-site skip authorization-check** argument and keyword are used.)

The **match certificate** command and **allow expired-certificate** keyword would be used at the central site if the router at a branch site had an expired certificate and it had to establish a tunnel to the central site to renew its certificate.

Trustpoint on the Central Site Router

crypto pki trustpoint VPN-GW
  enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
  serial-number none
  fqdn none
  ip-address none
  subject-name o=Home Office Inc, cn=Central VPN Gateway
  revocation-check crl

Trustpoint on the Branch 1 Site Router

Router#  **show crypto ca certificate**
Certificate
  Status: Available
  Certificate Serial Number: 2F62BE140000000000CA0
  Certificate Usage: General Purpose
  Issuer: cn=Central Certificate Authority
  o=Home Office Inc
  Subject:
  Name: Branch 1 Site
  cn=Branch 1 Site
  o=Home Office Inc
  CRL Distribution Points:
  http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
  start date: 00:43:26 GMT Sep 26 2003
  end date: 00:53:26 GMT Oct 3 2003
  renew date: 00:00:00 GMT Jan 1 1970
  Associated Trustpoints: home-office
CA Certificate
  Status: Available
  Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
  Certificate Usage: Signature
  Issuer: cn=Central Certificate Authority
  o=Home Office Inc
  Subject:
  cn=Central Certificate Authority
  o=Home Office Inc
  CRL Distribution Points:
  http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
  start date: 22:19:29 GMT Oct 31 2002
  end date: 22:27:27 GMT Oct 31 2017
  Associated Trustpoints: home-office

A certificate map is entered on the central site router.

Router#  **configure terminal**
Enter configuration commands, one per line. End with CNTL/Z.
Router (config)#  **crypto pki certificate map branch1 10**
Router (ca-certificate-map)#  **issuer-name co=Central Certificate Authority, ou=Home Office Inc**
!The above line wrapped but should be part of the line above it.
Router (ca-certificate-map)#  **subject-name eq cn=Brahcn 1 Site, o=home office inc**
Configuring Certificate Authorization and Revocation Settings Examples

The certificate map is added to the trustpoint.

```
Router (ca-certificate-map)# crypto pki trustpoint VPN-GW
Router (ca-trustpoint)# match certificate branch1 allow expired-certificate
Router (ca-trustpoint)# exit
Router (config) # exit
```

The configuration should be checked (most of the configuration is not shown).

```
Router# write term
! many lines left out
crypto pki trustpoint VPN-GW
  enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
  serial-number none
  fqdn none
  ip-address none
  subject-name o=Home Office Inc,cn=Central VPN Gateway
  revocation-check crl
  match certificate branch1 allow expired-certificate
  
  crypto pki certificate map central-site 10
  issuer-name cn = Central Certificate Authority, ou = Home Office Inc
  subject-name eq cn = central vpn gateway, o = home office inc
  ! many lines left out
```

The `match certificate` command and `branch1 allow expired-certificate` (argument and keyword) and the certificate map should be removed as soon as the branch router has a new certificate.

Configuring Certificate Authorization and Revocation Settings Examples

This section contains the following configuration examples that can be used when specifying a CRL cache control setting or certificate serial number session control:

### Configuring CRL Cache Control

The following example shows how to disable CRL caching for all CRLs associated with the CA1 trustpoint:

```
crypto pki trustpoint CA1
  enrollment url http://CA1:80
  ip-address FastEthernet0/0
  crl query ldap://ldap_CA1
  crl query ldap://ldap_CA1
  crl query ldap://ldap_CA1
  crl query ldap://ldap_CA1
  crl-cache none
```

The current CRL is still cached immediately after executing the example configuration shown above:

```
Router# show crypto pki crls
```

When the current CRL expires, a new CRL is then downloaded to the router at the next update. The `crl-cache none` command takes effect and all CRLs for the trustpoint are no longer cached; caching is disabled. You can verify that no CRL is cached by executing the `show crypto pki crls` command. No output will be shown because there are no CRLs cached.
The following example shows how to configure the maximum lifetime of 2 minutes for all CRLs associated with the CA1 trustpoint:

```plaintext
crypto pki trustpoint CA1
  enrollment url http://CA1:80
  ip-address FastEthernet0/0
  crl query ldap://ldap:CA1
  revocation-check crl
  crl-cache delete-after 2
```

The current CRL is still cached immediately after executing the example configuration above for setting the maximum lifetime of a CRL:

```plaintext
Router# show crypto pki crls
```

CRL Issuer Name:
- cn=Name Cert Manager,ou=pki,o=example.com,c=US
  LastUpdate: 18:57:42 GMT Nov 26 2005
  NextUpdate: 22:57:42 GMT Nov 26 2005
  Retrieved from CRL Distribution Point:
    ldap://ldap.example.com/CN=Name Cert Manager,O=example.com

When the current CRL expires, a new CRL is downloaded to the router at the next update and the `crl-cache delete-after` command takes effect. This newly cached CRL and all subsequent CRLs will be deleted after a maximum lifetime of 2 minutes.

You can verify that the CRL will be cached for 2 minutes by executing the `show crypto pki crls` command. Note that the NextUpdate time is 2 minutes after the LastUpdate time.

```plaintext
Router# show crypto pki crls
```

CRL Issuer Name:
- cn=Name Cert Manager,ou=pki,o=example.com,c=US
  LastUpdate: 22:57:42 GMT Nov 26 2005
  NextUpdate: 22:59:42 GMT Nov 26 2005
  Retrieved from CRL Distribution Point:
    ldap://ldap.example.com/CN=Name Cert Manager,O=example.com

### Configuring Certificate Serial Number Session Control

The following example shows the configuration of certificate serial number session control using a certificate map for the CA1 trustpoint:

```plaintext
crypto pki trustpoint CA1
  enrollment url http://CA1
  chain-validation stop
  crl query ldap://ldap_server
  revocation-check crl
  match certificate crl
  !
  crypto pki certificate map crl 10
  serial-number co 279d
```

**Note**

If the `match-criteria` value is set to `eq` (equal) instead of `co` (contains), the serial number must match the certificate map serial number exactly, including any spaces.

The following example shows the configuration of certificate serial number session control using AAA attributes. In this case, all valid certificates will be accepted if the certificate does not have the serial number "4ACA."

```plaintext
crypto pki trustpoint CA1
```
The server log shows that the certificate with the serial number “4ACA” was rejected. The certificate rejection is shown using exclamation points.

Dec 3 04:24:39.175: CRYPTO_PKI: Certificate received from 192.0.2.43 is bad: certificate invalid
Dec 3 04:24:39.175: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Main mode failed with peer at 192.0.2.43

Configuring Certificate Chain Validation Examples

This section contains the following configuration examples that can be used to specify the level of certificate chain processing for your device certificates:

Configuring Certificate Chain Validation from Peer to Root CA

In the following configuration example, all of the certificates will be validated--the peer, SubCA11, SubCA1, and RootCA certificates.

crypto pki trustpoint RootCA
  enrollment terminal
  chain-validation stop
Configuring Certificate Chain Validation from Peer to Subordinate CA

In the following configuration example, the following certificates will be validated—the peer and SubCA1 certificates.

```
crypto pki trustpoint RootCA
enrollment terminal
chain-validation stop
revocation-check none
rsakeypair RootCA
```
```
crypto pki trustpoint SubCA1
```
```
enrollment terminal
chain-validation continue RootCA
```
```
revocation-check none
rsakeypair SubCA1
```
```
crypto pki trustpoint SubCA11
```
```
enrollment terminal
chain-validation continue SubCA1
```
```
revocation-check none
rsakeypair SubCA11
```

Configuring Certificate Chain Validation Through a Gap

In the following configuration example, SubCA1 is not in the configured Cisco IOS hierarchy but is expected to have been supplied in the certificate chain presented by the peer.

If the peer supplies the SubCA1 certificate in the presented certificate chain, the following certificates will be validated—the peer, SubCA11, and SubCA1 certificates.

If the peer does not supply the SubCA1 certificate in the presented certificate chain, the chain validation will fail.

```
crypto pki trustpoint RootCA
```
```
enrollment terminal
chain-validation stop
revocation-check none
rsakeypair RootCA
```
```
crypto pki trustpoint SubCA11
```
```
enrollment terminal
chain-validation continue RootCA
```
```
revocation-check none
rsakeypair SubCA11
```

Configuring Certificate Servers for High Availability Example

The following example shows the configuration of SCTP and redundancy on the active and the standby certificate server, and activation of synchronization between them:
On the Active Router

ipc zone default
association 1
no shutdown
protocol sctp
local-port 5000
local-ip 10.0.0.1
exit
remote-port 5000
remote-ip 10.0.0.2

On the Standby Router

ipc zone default
association 1
no shutdown
protocol sctp
local-port 5000
local-ip 10.0.0.2
exit
remote-port 5000
remote-ip 10.0.0.1

On the Active Router

redundancy inter-device
scheme standby SB
interface GigabitEthernet0/1
ip address 10.0.0.1 255.255.255.0
no ip route-cache cef
no ip route-cache
standby 0 ip 10.0.0.3
standby 0 priority 50
standby 0 name SB
standby delay min 30 reload 60

On the Standby Router

redundancy inter-device
scheme standby SB
interface GigabitEthernet0/1
ip address 10.0.0.2 255.255.255.0
no ip route-cache cef
no ip route-cache
standby 0 ip 10.0.0.3
standby 0 priority 50
standby 0 name SB
standby delay min 30 reload 60

On the Active Router

crypto pki server mycertsaver
crypto pki server mycertsaver redundancy
### Additional References

#### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples</td>
<td><em>Cisco IOS Security Command Reference</em></td>
</tr>
<tr>
<td>Overview of PKI, including RSA keys, certificate enrollment, and CAs</td>
<td>“Cisco IOS PKI Overview: Understanding and Planning a PKI” module</td>
</tr>
<tr>
<td>RSA key generation and deployment</td>
<td>“Deploying RSA Keys Within a PKI” module</td>
</tr>
<tr>
<td>Certificate enrollment: supported methods, enrollment profiles, configuration tasks</td>
<td>“Configuring Certificate Enrollment for a PKI” module</td>
</tr>
<tr>
<td>Cisco IOS certificate server overview information and configuration tasks</td>
<td>“Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment” module</td>
</tr>
<tr>
<td>Recommended cryptographic algorithms</td>
<td>Next Generation Encryption</td>
</tr>
</tbody>
</table>

#### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>
Table 3: Feature Information for PKI Certificate Authorization and Revocation

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Cache Control Enhancements for Certification Revocation Lists | 12.4(9)T   | This feature provides users the ability to disable CRL caching or to specify the maximum lifetime for which a CRL will be cached in router memory. It also provides functionality to configure certificate serial number session control. The following sections provide information about this feature:  
- What Is a CRL, on page 41  
- Configuring Certificate Authorization and Revocation Settings, on page 53  
- Configuring SCTP on the Active and Standby Certificate Servers, on page 66  
The following commands were introduced or modified by this feature: crl-cache delete-after, crl-cache none, crypto pki certificate map |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Certificate-Complete Chain Validation            | 12.4(6)T  | This feature provides users the ability to configure the level to which a certificate chain is processed on all certificates including subordinate CA certificates. The following sections provide information about this feature:  
  - PKI Certificate Chain Validation, on page 44  
  - Configuring Certificate Chain Validation, on page 61  
  - Configuring Certificate Chain Validation Examples, on page 80  
  The following command was introduced by this feature:  
    chain-validation |
| OCSP - Server Certification from Alternate Hierarchy | 12.4(6)T  | This feature provides users with the flexibility to specify multiple OCSP servers, either per client certificate or per group of client certificates, and provides the capability for OCSP server validation based on external CA certificates or self-signed certificates. The following sections provide information about this feature:  
  - What Is OCSP, on page 42  
  - Configuring Certificate Authorization and Revocation Settings, on page 53  
  The following command was introduced by this feature:  
    match certificate override ocs
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Optional OCSP Nonce      | 12.2(33)SR 12.4(4)T | This feature provides users with the ability to configure the sending of a nonce, or unique identifier for an OCSP request, during OCSP communications.  
The following sections provide information about this feature:
  - What Is OCSP, on page 42
  - Configuring a Revocation Mechanism for PKI Certificate Status Checking, on page 50
  - Disabling Nonces in Communications with the OCSP Server Example, on page 74 |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Certificate Security Attribute-Based Access Control | 12.2(15)T 1 | Under the IPsec protocol, CA interoperability permits Cisco IOS devices and a CA to communicate so that the Cisco IOS device can obtain and use digital certificates from the CA. Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action. This feature adds fields to the certificate that allow specifying an ACL, creating a certificate-based ACL. The following sections provide information about this feature:  
  - When to Use Certificate-Based ACLs for Authorization or Revocation, on page 43  
  - Configuring Certificate Authorization and Revocation Settings, on page 53  

The following commands were introduced or modified by this feature: `crypto pki certificate map`, `crypto pki trustpoint match certificate`
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Certificate Status Protocol (OCSP)</td>
<td>12.3(2)T</td>
<td>This feature allows users to enable OCSP instead of CRLs to check certificate status. Unlike CRLs, which provide only periodic certificate status, OCSP can provide timely information regarding the status of a certificate. The following sections provide information about this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CRLs or OCSP Server Choosing a Certificate Revocation Mechanism, on page 41 &lt;br&gt; • Configuring a Revocation Mechanism for PKI Certificate Status Checking, on page 50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following commands were introduced by this feature: <code>ocsp url</code>, <code>revocation-check</code></td>
</tr>
<tr>
<td>PKI AAA Authorization Using the Entire Subject Name</td>
<td>12.3(11)T</td>
<td>This feature provides users with the ability to query the AAA server using the entire subject name from the certificate as a unique AAA username. The following sections provide information about this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Attribute-Value Pairs for PKI and AAA Server Integration, on page 40 &lt;br&gt; • Configuring PKI Integration with a AAA Server, on page 46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following command was modified by this feature: <code>authorization username</code></td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| PKI Integration with AAA Server | 12.3(1)  | This feature provides additional scalability for authorization by generating a AAA username from the certificate presented by the peer. A AAA server is queried to determine whether the certificate is authorized for use by the internal component. The authorization is indicated by a component-specified label that must be present in the AV pair for the user. The following sections provide information about this feature:  
  - PKI and AAA Server Integration for Certificate Status, on page 39  
  - Configuring PKI Integration with a AAA Server, on page 46  
  The following commands were introduced by this feature:  
    authorization list, authorization username |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| PKI: Query Multiple Servers During Certificate Revocation Check | 12.3(7)T | This feature introduces the ability for Cisco IOS software to make multiple attempts to retrieve the CRL, allowing operations to continue when a particular server is not available. In addition, the ability to override the CDPs in a certificate with a manually configured CDP has been introduced. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate’s CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP.

The following sections provide information about this feature:

- **Querying All CDPs During Revocation Check**, on page 41
- **Manually Overriding CDPs in a Certificate**, on page 54

The following command was introduced by this feature: `match certificate override cdp`
## Feature Information for Certificate Authorization and Revocation

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Using Certificate ACLs to Ignore Revocation Check and Expired Certificates | 12.3(4)T | This feature allows a certificate that meets specified criteria to be accepted regardless of the validity period of the certificate, or if the certificate meets the specified criteria, revocation checking does not have to be performed. Certificate ACLs are used to specify the criteria that the certificate must meet to be accepted or to avoid revocation checking. In addition, if AAA communication is protected by a certificate, this feature provides for the AAA checking of the certificate to be ignored. The following sections provide information about this feature:
- Ignore Revocation Checks Using a Certificate-Based ACL, on page 43
- Configuring Certificate-Based ACLs to Ignore Revocation Checks, on page 53

The following command was modified by this feature: **match certificate** |

| Query Mode Definition Per Trustpoint | Cisco IOS XE Release 2.1 | This feature was introduced on the Cisco ASR 1000 series routers. |

<p>| PKI High Availability | 15.0(1)M | The following commands were introduced or modified: <strong>crypto pki server, crypto pki server start, crypto pki server stop, crypto pki trustpoint, crypto key generate rsa, crypto key import pem, crypto key move rsa, show crypto key mypubkey rsa.</strong> |</p>
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI IPv6 Support for VPN Solutions</td>
<td>15.2(1)T</td>
<td>The <code>enrollment url (ca-trustpoint)</code> command was modified to specify an IPv6 address in the CA URL.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The <code>ocsp url</code> command was modified to specify the IPv6 address in a URL for the OCSP server.</td>
</tr>
</tbody>
</table>
Configuring Certificate Enrollment for a PKI

This module describes the different methods available for certificate enrollment and how to set up each method for a participating PKI peer. Certificate enrollment, which is the process of obtaining a certificate from a certification authority (CA), occurs between the end host that requests the certificate and the CA. Each peer that participates in the public key infrastructure (PKI) must enroll with a CA.

Note

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

• Finding Feature Information, page 93
• Prerequisites for PKI Certificate Enrollment, page 94
• Information About Certificate Enrollment for a PKI, page 94
• How to Configure Certificate Enrollment for a PKI, page 98
• Configuration Examples for PKI Certificate Enrollment Requests, page 123
• Additional References, page 130
• Feature Information for PKI Certificate Enrollment, page 132

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.
Prerequisites for PKI Certificate Enrollment

Before configuring peers for certificate enrollment, you should have the following items:

- A generated Rivest, Shamir, and Adelman (RSA) key pair to enroll and a PKI in which to enroll.
- An authenticated CA.
- Familiarity with the module “Cisco IOS PKI Overview: Understanding and Planning a PKI.”

Note
As of Cisco IOS Release 12.3(7)T, all commands that begin with "crypto ca" have been changed to begin with "crypto pki." Although the router will still accept crypto ca commands, all output will be displayed crypto pki.

Information About Certificate Enrollment for a PKI

What Are CAs

A CA is an entity that issues digital certificates that other parties can use. It is an example of a trusted third party. CAs are characteristic of many PKI schemes.

A CA manages certificate requests and issues certificates to participating network devices. These services provide centralized key management for the participating devices to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

You can use the Cisco IOS certificate server or a CA provided by a third-party CA vendor.

Framework for Multiple CAs

A PKI can be set up in a hierarchical framework to support multiple CAs. At the top of the hierarchy is a root CA, which holds a self-signed certificate. The trust within the entire hierarchy is derived from the RSA key pair of the root CA. The subordinate CAs within the hierarchy can be enrolled with either the root CA or with another subordinate CA. Multiple tiers of CAs are configured by either the root CA or with another subordinate CA. Within a hierarchical PKI, all enrolled peers can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA.

When to Use Multiple CAs

Multiple CAs provide users with added flexibility and reliability. For example, subordinate CAs can be placed in branch offices while the root CA is at the office headquarters. Also, different granting policies can be implemented per CA, so you can set up one CA to automatically grant certificate requests while another CA within the hierarchy requires each certificate request to be manually granted.

Scenarios in which at least a two-tier CA is recommended are as follows:
• Large and very active networks in which a large number of certificates are revoked and reissued. A multiple tier CA helps to control the size of the certificate revocation lists (CRLs).

• When online enrollment protocols are used, the root CA can be kept offline except to issue subordinate CA certificates. This scenario provides added security for the root CA.

**Authentication of the CA**

The certificate of the CA must be authenticated before the device will be issued its own certificate and before certificate enrollment can occur. Authentication of the CA typically occurs only when you initially configure PKI support at your router. To authenticate the CA, issue the `crypto pki authenticate` command, which authenticates the CA to your router by obtaining the self-signed certificate of the CA that contains the public key of the CA.

**Authentication via the fingerprint Command**

Cisco IOS Release 12.3(12) and later releases allow you to issue the `fingerprint` command to preenter a fingerprint that can be matched against the fingerprint of a CA certificate during authentication.

If a fingerprint is not preentered for a trustpoint, and if the authentication request is interactive, you must verify the fingerprint that is displayed during authentication of the CA certificate. If the authentication request is noninteractive, the certificate will be rejected without a preentered fingerprint.

**Note**

If the authentication request is made using the command-line interface (CLI), the request is an interactive request. If the authentication request is made using HTTP or another management tool, the request is a noninteractive request.

**Supported Certificate Enrollment Methods**

Cisco IOS software supports the following methods to obtain a certificate from a CA:

• Simple Certificate Enrollment Protocol (SCEP)—A Cisco-developed enrollment protocol that uses HTTP to communicate with the CA or registration authority (RA). SCEP is the most commonly used method for sending and receiving requests and certificates.

**Note**

To take advantage of automated certificate and key rollover functionality, you must be running a CA that supports rollover and SCEP must be used as your client enrollment method. If you are running a Cisco IOS CA, you must be running Cisco IOS Release 12.4(2)T or a later release for rollover support.

• PKCS12—The router imports certificates in PKCS12 format from an external server.

• IOS File System (IFS)—The router uses any file system that is supported by Cisco IOS software (such as TFTP, FTP, flash, and NVRAM) to send a certificate request and to receive the issued certificate. Users may enable IFS certificate enrollment when their CA does not support SCEP.
Prior to Cisco IOS Release 12.3(4)T, only the TFTP file system was supported within IFS.

- Manual cut-and-paste: The router displays the certificate request on the console terminal, allowing the user to enter the issued certificate on the console terminal. A user may manually cut-and-paste certificate requests and certificates when there is no network connection between the router and CA.
- Enrollment profiles: The router sends HTTP-based enrollment requests directly to the CA server instead of to the RA-mode certificate server (CS). Enrollment profiles can be used if a CA server does not support SCEP.
- Self-signed certificate enrollment for a trustpoint: The secure HTTP (HTTPS) server generates a self-signed certificate that is to be used during the secure socket layer (SSL) handshake, establishing a secure connection between the HTTPS server and the client. The self-signed certificate is then saved in the router’s startup configuration (NVRAM). The saved, self-signed certificate can then be used for future SSL handshakes, eliminating the user intervention that was necessary to accept the certificate every time the router reloaded.

To take advantage of autoenrollment and autoreenrollment, do not use either TFTP or manual cut-and-paste enrollment as your enrollment method. Both TFTP and manual cut-and-paste enrollment methods are manual enrollment processes, requiring user input.

**Cisco IOS Suite-B Support for Certificate Enrollment for a PKI**

Suite-B requirements comprise of four user interface suites of cryptographic algorithms for use with IKE and IPSec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm.

Suite-B adds the following support for the certificate enrollment for a PKI:

- Elliptic Curve Digital Signature Algorithm (ECDSA) (256-bit and 384-bit curves) is used for the signature operation within X.509 certificates.
- PKI support for validation of for X.509 certificates using ECDSA signatures.
- PKI support for generating certificate requests using ECDSA signatures and for importing the issued certificates into IOS.

See the Configuring Security for VPNs with IPsec feature module for more detailed information about Cisco IOS Suite-B support.

**Registration Authorities**

A Cisco IOS certificate server can be configured to run in RA mode. An RA offloads authentication and authorization responsibilities from a CA. When the RA receives a SCEP or manual enrollment request, the administrator can either reject or grant it on the basis of local policy. If the request is granted, it will be forwarded to the issuing CA, and the CA can be configured to automatically generate the certificate and return it to the RA. The client can later retrieve the granted certificate from the RA.
Automatic Certificate Enrollment

Automatic certificate enrollment allows the CA client to automatically request a certificate from its CA server. This automatic router request eliminates the need for operator intervention when the enrollment request is sent to the CA server. Automatic enrollment is performed on startup for any trustpoint CA that is configured and that does not have a valid client certificate. When the certificate expires, a new certificate is automatically requested.

Note

When automatic enrollment is configured, clients automatically request client certificates. The CA server performs its own authorization checks; if these checks include a policy to automatically issue certificates, all clients will automatically receive certificates, which is not very secure. Thus, automatic certificate enrollment should be combined with additional authentication and authorization mechanisms (such as Secure Device Provisioning (SDP), leveraging existing certificates, and one-time passwords).

Automated Client Certificate and Key Rollover

By default, the automatic certificate enrollment function requests a new client certificate and keys from the CS before the client’s current certificate expires. Certificate and key rollover allows the certificate renewal rollover request to be made before the certificate expires by retaining the current key and certificate until the new, or rollover, certificate is available. After a specified amount of time, the rollover certificate and keys will become the active certificate and keys. The expired certificate and keys are immediately deleted upon rollover and removed from the certificate chain and CRL.

The setup for automatic rollover is twofold: CA clients must be automatically enrolled and the client’s CAs must be automatically enrolled and have the auto-rollover command enabled. For more information on configuring your CA servers for automatic certificate rollover see the section "Automatic CA Certificate and Key Rollover" in the chapter "Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment " of the Public Key Infrastructure Configuration Guide.

An optional renewal percentage parameter can be used with the auto-enroll command to allow a new certificate to be requested when a specified percentage of the lifetime of the certificate has passed. For example, if the renewal percentage is configured as 90 and the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires. In order for automatic rollover to occur, the renewal percentage must be less than 100. The specified percent value must not be less than 10. If a client certificate is issued for less than the configured validity period due to the impending expiration of the CA certificate, the rollover certificate will be issued for the balance of that period. A minimum of 10 percent of the configured validity period, with an absolute minimum of 3 minutes, is required to allow rollover enough time to function.

Tip

If CA autoenrollment is not enabled, you may manually initiate rollover on an existing client with the crypto pki enroll command if the expiration time of the current client certificate is equal to or greater than the expiration time of the corresponding CA certificate. The client will initiate the rollover process, which occurs only if the server is configured for automated rollover and has an available rollover server certificate.
Certificate Enrollment Profiles

Certificate enrollment profiles allow users to specify certificate authentication, enrollment, and reenrollment parameters when prompted. The values for these parameters are referenced by two templates that make up the profile. One template contains parameters for the HTTP request that is sent to the CA server to obtain the certificate of the CA (also known as certificate authentication); the other template contains parameters for the HTTP request that is sent to the CA for certificate enrollment.

Configuring two templates enables users to specify different URLs or methods for certificate authentication and enrollment; for example, authentication (getting the certificate of the CA) can be performed via TFTP (using the **authentication url** command) and enrollment can be performed manually (using the **enrollment terminal** command).

Prior to Cisco IOS Release 12.3(11)T, certificate requests could be sent only in a PKCS10 format; however, an additional parameter was added to the profile, allowing users to specify the PKCS7 format for certificate renewal requests.

A single enrollment profile can have up to three separate sections for each task—certificate authentication, enrollment, and reenrollment.

How to Configure Certificate Enrollment for a PKI

This section contains the following enrollment option procedures. If you configure enrollment or autoenrollment (the first task), you cannot configure manual certificate enrollment. Also, if you configure TFTP or manual cut-and-paste certificate enrollment, you cannot configure autoenrollment, autoreenrollment, an enrollment profile, nor can you utilize the automated CA certificate rollover capability.

Configuring Certificate Enrollment or Autoenrollment

Perform this task to configure certificate enrollment or autoenrollment for clients participating in your PKI.

Before You Begin

Before configuring automatic certificate enrollment requests, you should ensure that all necessary enrollment information is configured.

Prerequisites for Enabling Automated Client Certificate and Key Rollover

CA client support for certificate rollover is automatically enabled when using autoenrollment. For automatic CA certificate rollover to run successfully, the following prerequisites are applicable:

- Your network devices must support shadow PKI.
• Your clients must be running Cisco IOS Release 12.4(2)T or a later release.

• The client’s CS must support automatic rollover. See the section "Automatic CA Certificate and Key Rollover" in the chapter "Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" of the Public Key Infrastructure Configuration Guide for more information on CA server automatic rollover configuration.

**Prerequisites for Specifying Autoenrollment Initial Key Generation Location**

To specify the location of the autoenrollment initial key generation, you must be running Cisco IOS Release 12.4(11)T or a later release.

---

**Note**

**RSA Key Pair Restriction for Autoenrollment**

Trustpoints configured to generate a new key pair using the `regenerate` command or the `regenerate` keyword of the `auto-enroll` command must not share key pairs with other trustpoints. To give each trustpoint its own key pair, use the `rsakeypair` command in ca-trustpoint configuration mode. Sharing key pairs among regenerating trustpoints is not supported and will cause loss of service on some of the trustpoints because of key and certificate mismatches.

**Restrictions for Automated Client Certificate and Key Rollover**

In order for clients to run automatic CA certificate rollover successfully, the following restrictions are applicable:

- SCEP must be used to support rollover. Any device that enrolls with the PKI using an alternative to SCEP as the certificate management protocol or mechanism (such as enrollment profiles, manual enrollment, or TFTP enrollment) will not be able to take advantage of the rollover functionality provided by SCEP.

- If the configuration cannot be saved to the startup configuration after a shadow certificate is generated, rollover will not occur.

---

**Note**

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.
## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment [mode | retry period minutes | retry count number] url url [pem]**
5. **eckeypair** *label*
6. **subject-name [x.500-name]**
7. **vrf** *vrf-name*
8. **ip-address** {ip-address | interface | none}
9. **serial-number [none]**
10. **auto-enroll [percent] [regenerate]**
11. **usage** *method1 [method2 [method3]]*
12. **password** *string*
13. **rsakeypair** {key-label key-size encryption-key-size}
14. **fingerprint** *ca-fingerprint*
15. **on** *devicename :
16. **exit**
17. **crypto pki authenticate** *name*
18. **exit**
19. **copy system:running-config nvram:startup-config**
20. **show crypto pki certificates**

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>enable</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>configure terminal</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>crypto pki trustpoint</strong> <em>name</em></td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

Command or Action | Purpose
--- | ---
**Step 4** enrollment [mode | retry period minutes | retry count number] url [pem] | Specifies the URL of the CA on which your router should send certificate requests.
  - **mode** -- Specifies RA mode if your CA system provides an RA.
  - **retry period minutes** -- Specifies the wait period between certificate request retries. The default is 1 minute between retries.
  - **retry count number** -- Specifies the number of times a router will resend a certificate request when it does not receive a response from the previous request. (Specify from 1 to 100 retries.)
  - **url** -- URL of the file system where your router should send certificate requests. An IPv6 address can be added in the URL enclosed in brackets. For example: http://[2001:DB8:1:1::1]:80. For more enrollment method options, see the enrollment url (ca-trustpoint) command page.
  - **pem** -- Adds privacy-enhanced mail (PEM) boundaries to the certificate request.

**Note** An enrollment method other than TFTP or manual cut-and-paste must be configured to support autoenrollment.

**Step 5** eckeypair label | (Optional) Configures the trustpoint to use an Elliptic Curve (EC) key on which certificate requests are generated using ECDSA signatures. The label argument specifies the EC key label that is configured using the crypto key generate rsa or crypto key generate ec keysize command in global configuration mode. See the Configuring Internet Key Exchange for IPsec VPNs feature module for more information.

**Note** If an ECDSA signed certificate is imported without a trustpoint configuration, then the label defaults to the FQDN value.

**Step 6** subject-name [x.500-name] | (Optional) Specifies the requested subject name that will be used in the certificate request.

  - **x.500-name** -- If it is not specified, the fully qualified domain name (FQDN), which is the default subject name, will be used.

**Step 7** vrf vrf-name | (Optional) Specifies the the VRF instance in the public key infrastructure (PKI) trustpoint to be used for enrollment, certificate revocation list (CRL) retrieval, and online certificate status protocol (OCSP) status.

**Step 8** ip-address {ip-address | interface | none} | (Optional) Includes the IP address of the specified interface in the certificate request.

  - Issue the **ip-address** argument to specify either an IPv4 or IPv6 address.
  - Issue the **interface** argument to specify an interface on the router.
  - Issue the **none** keyword if no IP address should be included.
### Purpose

If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.

### Step 9

**serial-number [none]**

(Optional) Specifies the router serial number in the certificate request, unless the `none` keyword is issued.

- Issue the `none` keyword to specify that a serial number will not be included in the certificate request.

**Example:**

Router(ca-trustpoint)# serial-number

### Step 10

**auto-enroll [percent] [regenerate]**

(Optional) Enables autoenrollment, allowing the client to automatically request a rollover certificate from the CA.

- If autoenrollment is not enabled, the client must be manually re-enrolled in your PKI upon certificate expiration.
- By default, only the Domain Name System (DNS) name of the router is included in the certificate.
- Use the `percent` argument to specify that a new certificate will be requested after the percentage of the lifetime of the current certificate is reached.
- Use the `regenerate` keyword to generate a new key for the certificate even if a named key already exists.

**Note** If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: “! RSA key pair associated with trustpoint is exportable.”

**Note** It is recommended that a new key pair be generated for security reasons.

**Example:**

Router(ca-trustpoint)# auto-enroll regenerate

### Step 11

**usage method1 [method2 [method3]]**

(Optional) Specifies the intended use for the certificate.

- Available options are `ike`, `ssl-client`, and `ssl-server`; the default is `ike`.

**Example:**

Router(ca-trustpoint)# usage ssl-client

### Step 12

**password string**

(Optional) Specifies the revocation password for the certificate.

- If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint.

**Note** When SCEP is used, this password can be used to authorize the certificate request—often via a one-time password or similar mechanism.

**Example:**

Router(ca-trustpoint)# password string

### Step 13

**rsakeypair key-label key-size encryption-key-size**

(Optional) Specifies which key pair to associate with the certificate.

- A key pair with the `key-label` argument will be generated during enrollment if it does not already exist or if the `auto-enroll regenerate` command was issued.
- Specify the `key-size` argument for generating the key, and specify the `encryption-key-size` argument to request separate encryption, signature
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>keys, and certificates. The key-size and encryption-key-size must be the same size. Length of less than 2048 is not recommended.</td>
<td><strong>Note</strong> If this command is not enabled, the FQDN key pair is used.</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td><strong>fingerprint ca-fingerprint</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router (ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E</td>
</tr>
<tr>
<td>(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication.</td>
<td><strong>Note</strong> If the fingerprint is not provided and authentication of the CA certificate is interactive, the fingerprint will be displayed for verification.</td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td><strong>on devicename :</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router (ca-trustpoint)# on usbtoken0:</td>
</tr>
<tr>
<td>(Optional) Specifies that RSA keys will be created on the specified device upon autoenrollment initial key generation.</td>
<td></td>
</tr>
<tr>
<td>• Devices that may be specified include NVRAM, local disks, and Universal Serial Bus (USB) tokens. USB tokens may be used as cryptographic devices in addition to a storage device. Using a USB token as a cryptographic device allows RSA operations such as key generation, signing, and authentication to be performed on the token.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td><strong>exit</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router (ca-trustpoint)# exit</td>
</tr>
<tr>
<td>Exits ca-trustpoint configuration mode and returns to global configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 17</strong></td>
<td><strong>crypto pki authenticate name</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router (config)# crypto pki authenticate mytp</td>
</tr>
<tr>
<td>Retrieves the CA certificate and authenticates it.</td>
<td></td>
</tr>
<tr>
<td>• Check the certificate fingerprint if prompted.</td>
<td><strong>Note</strong> This command is optional if the CA certificate is already loaded into the configuration.</td>
</tr>
<tr>
<td><strong>Step 18</strong></td>
<td><strong>exit</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router (config)# exit</td>
</tr>
<tr>
<td>Exits global configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 19</strong></td>
<td><strong>copy system:running-config nvram:startup-config</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# copy system:running-config nvram:startup-config</td>
</tr>
<tr>
<td>(Optional) Copies the running configuration to the NVRAM startup configuration.</td>
<td><strong>Note</strong> Autoenrollment will not update NVRAM if the running configuration has been modified but not written to NVRAM.</td>
</tr>
</tbody>
</table>
### Configuring Manual Certificate Enrollment

Manual certificate enrollment can be set up via TFTP or the manual cut-and-paste method. Both options can be used if your CA does not support SCEP or if a network connection between the router and CA is not possible. Perform one of the following tasks to set up manual certificate enrollment:

#### PEM-Formatted Files for Certificate Enrollment Request

Using PEM-formatted files for certificate requests can be helpful for customers who are using terminal or profile-based enrollment to request certificates from their CA server. Customers using PEM-formatted files can directly use existing certificates on their routers.

#### Restrictions for Manual Certificate Enrollment

**SCEP Restriction**

We do not recommend switching URLs if SCEP is used; that is, if the enrollment URL is “http://myca,” do not change the enrollment URL after getting the CA certificate and before enrolling the certificate. A user can switch between TFTP and manual cut-and-paste.

**Key Regeneration Restriction**

Do not regenerate the keys manually using the `crypto key generate` command; key regeneration will occur when the `crypto pki enroll` command is issued if the `regenerate` keyword is specified.

#### Configuring Cut-and-Paste Certificate Enrollment

Perform this task to configure cut-and-paste certificate enrollment. This task helps you to configure manual certificate enrollment via the cut-and-paste method for peers participating in your PKI.

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 20</strong></td>
<td></td>
</tr>
<tr>
<td><code>show crypto pki certificates</code></td>
<td>(Optional) Displays information about your certificates, including any rollover certificates.</td>
</tr>
</tbody>
</table>

---

**Example:**

```
Router# show crypto pki certificates
```
SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. enrollment terminal pem
5. fingerprint ca-fingerprint
6. exit
7. crypto pki authenticate name
8. crypto pki enroll name
9. crypto pki import name certificate
10. exit
11. show crypto pki certificates

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>crypto pki trustpoint name</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# crypto pki trustpoint mytp</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>enrollment terminal pem</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(ca-trustpoint)# enrollment terminal</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 5</th>
<th>fingerprint <em>ca-fingerprint</em></th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> If the fingerprint is not provided, it will be displayed for verification.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>exit</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exits ca-trustpoint configuration mode and returns to global configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th><code>crypto pki authenticate name</code></th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Retrieves the CA certificate and authenticates it.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th><code>crypto pki enroll name</code></th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Generates certificate request and displays the request for copying and pasting into the certificate server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• You are prompted for enrollment information, such as whether to include the router FQDN and IP address in the certificate request. You are also given the choice about displaying the certificate request to the console terminal.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The base-64 encoded certificate with or without PEM headers as requested is displayed.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th><code>crypto pki import name certificate</code></th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Imports a certificate manually at the console terminal (pasting).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The base-64 encoded certificate is accepted from the console terminal and inserted into the internal certificate database.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> You must enter this command twice if usage keys, a signature key, and an encryption key are used. The first time the command is entered, one of the certificates is pasted into the router. The second time the command is entered, the other certificate is pasted into the router. It does not matter which certificate is pasted first. Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If this applies to the certificate authority you are using, import the general purpose certificate. The router will not use one of the two key pairs generated.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th>exit</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exits global configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring TFTP Certificate Enrollment

Perform this task to configure TFTP certificate enrollment. This task helps you to configure manual certificate enrollment using a TFTP server.

**Before You Begin**

- You must know the correct URL to use if you are configuring certificate enrollment via TFTP.
- The router must be able to write a file to the TFTP server for the `crypto pki enroll` command.
- If you are using a file specification with the `enrollment` command, the file must contain the CA certificate either in binary format or in base-64 encoded.
- You must know if your CA ignores key usage information in a certificate request and issues only a general purpose usage certificate.

**Caution**

Some TFTP servers require that the file must exist on the server before it can be written. Most TFTP servers require files that can be written over. This requirement may pose a risk because any router or other device may write or overwrite the certificate request; thus, the replacement certificate request will not be used by the CA administrator, who must first check the enrollment request fingerprint before granting the certificate request.
SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. enrollment [mode] [retry period minutes] [retry count number] url [url [pem]]
5. fingerprint ca-fingerprint
6. exit
7. crypto pki authenticate name
8. crypto pki enroll name
9. crypto pki import name certificate
10. exit
11. show crypto pki certificates

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 1 enable     | Enables privileged EXEC mode.  
| Example:          | • Enter your password if prompted. |
| Router> enable    |         |

| Step 2 configure terminal | Enters global configuration mode.  
| Example:                  | |
| Router# configure terminal|         |

| Step 3 crypto pki trustpoint name | Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.  
| Example:                           | |
| Router(config)# crypto pki trustpoint mytp |         |

| Step 4 enrollment [mode] [retry period minutes] [retry count number] url [pem] | Specifies TFTP as the enrollment method to send the enrollment request and to retrieve the CA certificate and router certificate and any optional parameters.  
| Note: For TFTP enrollment, the URL must be configured as a TFTP URL, tftp://example_tftp_url.  
| Example:                                      | • An optional file specification filename may be included in the TFTP URL. If the file specification is not included, the FQDN will be used. If the file specification is included, the router will append the extension ".ca" to the specified filename. |
| Router(ca-trustpoint)# enrollment url tftp://certserver/file_specification |         |
### Step 5

**fingerprint ca-fingerprint**

(Optional) Specifies the fingerprint of the CA certificate received via an out-of-band method from the CA administrator.

**Example:**

```
Router(ca-trustpoint)# fingerprint
12EF53FA 355CD23E 12EF53FA 355CD23E
```

**Note**

If the fingerprint is not provided, it will be displayed for verification.

---

### Step 6

**exit**

Exits ca-trustpoint configuration mode and returns to global configuration mode.

**Example:**

```
Router(ca-trustpoint)# exit
```

---

### Step 7

**crypto pki authenticate name**

Retrieves the CA certificate and authenticates it from the specified TFTP server.

**Example:**

```
Router(config)# crypto pki authenticate mytp
```

---

### Step 8

**crypto pki enroll name**

Generates certificate request and writes the request out to the TFTP server.

- You are prompted for enrollment information, such as whether to include the router FQDN and IP address in the certificate request. You are queried about whether to display the certificate request to the console terminal.

- The filename to be written is appended with the extension ".req". For usage keys, a signature key and an encryption key, two requests are generated and sent. The usage key request filenames are appended with the extensions "-sign.req" and "-encr.req", respectively.

**Example:**

```
Router(config)# crypto pki enroll mytp
```

---

### Step 9

**crypto pki import name certificate**

Imports a certificate via TFTP at the console terminal, which retrieves the granted certificate.

- The router will attempt to retrieve the granted certificate via TFTP using the same filename used to send the request, except the extension is changed from ".req" to ".crt". For usage key certificates, the extensions ".sign.crt" and ".encr.crt" are used.

- The router will parse the received files, verify the certificates, and insert the certificates into the internal certificate database on the router.

**Example:**

```
Router(config)# crypto pki import mytp certificate
```

**Note**

Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If your CA ignores the usage key information in the certificate request, only import the general purpose certificate. The router will not use one of the two key pairs generated.
<table>
<thead>
<tr>
<th>Step 10</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>exit</td>
<td>Exits global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config)# exit

<table>
<thead>
<tr>
<th>Step 11</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>show crypto pki certificates</td>
<td>(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.</td>
</tr>
</tbody>
</table>

**Example:**

Router# show crypto pki certificates

---

**Certifying a URL Link for Secure Communication with a Trend Micro Server**

Perform this task to certify a link used in URL filtering that allows secure communication with a Trend Micro Server.

**Note**

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.
SUMMARY STEPS

1. `enable`
2. `clock set hh : mm : ss date month year`
3. `configure terminal`
4. `clock timezone zone hours-offset [minutes-offset ]`
5. `ip http server`
6. `hostname name`
7. `ip domain-name name`
8. `crypto key generate rsa general-keys modulus modulus-size`
9. `crypto pki trustpoint name`
10. `enrollment terminal`
11. `crypto ca authenticate name`
12. Copy the following block of text containing the base 64 encoded CA certificate and paste it at the prompt.
13. Enter `yes` to accept this certificate.
14. `serial-number`
15. `revocation-check none`
16. `end`
17. `trm register`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td>• Enter your password if prompted.</td>
</tr>
</tbody>
</table>
| Example:
  Router> enable | |
| **Step 2** | Sets the clock on the router. |
| `clock set hh : mm : ss date month year` | |
| Example:
  Router# clock set 23:22:00 22 Dec 2009 | |
| **Step 3** | Enters global configuration mode. |
| `configure terminal` | |
| Example:
  Router# configure terminal | |
<p>| <strong>Step 4</strong> | Sets the time zone. |
| <code>clock timezone zone hours-offset [minutes-offset ]</code> | • The <code>zone</code> argument is the name of the time zone (typically a standard acronym). The <code>hours-offset</code> argument is the number of hours the time |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong></td>
<td>Enables the HTTP server.</td>
</tr>
<tr>
<td>ip http server</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Configures the hostname of the router.</td>
</tr>
<tr>
<td>hostname name</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Defines the domain name for the router.</td>
</tr>
<tr>
<td>ip domain-name name</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Generates the crypto keys.</td>
</tr>
<tr>
<td>crypto key generate rsa general-keys modulus modulus-size</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Declares the CA that your router should use and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td>crypto pki trustpoint name</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Specifies the manual cut-and-paste certificate enrollment method.</td>
</tr>
<tr>
<td>enrollment terminal</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**
Router(config)# clock timezone PST -08

**Note**
The `minutes-offset` argument of the `clock timezone` command is available for those cases where a local time zone is a percentage of an hour different from UTC or Greenwich Mean Time (GMT). For example, the time zone for some sections of Atlantic Canada (AST) is UTC-3.5. In this case, the necessary command would be `clock timezone AST -3 30`.

**Example:**
Router(config)# ip http server

**Example:**
Router(config)#hostname hostname1

**Example:**
Router(config)# ip domain-name example.com

**Step 8**
- The `general-keys` keyword specifies that a general purpose key pair is generated, which is the default.
- The `modulus` keyword and `modulus-size` argument specify the IP size of the key modulus. By default, the modulus of a CA key is 1024 bits. When generating RSA keys, you will be prompted to enter a modulus length. A longer modulus could offer stronger security but takes longer to generate and to use. A length of less than 2048 is not recommended.

**Note**
The name for the general keys that are generated are based on the domain name that is configured in Step 7. For example, the keys will be called “example.com.”

**Step 9**
- Effective with Cisco IOS Release 12.3(8)T, the `crypto pki trustpoint` command replaced the `crypto ca trustpoint` command.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>• The certificate request will be displayed on the console terminal so that you may manually copy (or cut).</td>
</tr>
<tr>
<td>Router(ca-trustpoint)# enrollment terminal</td>
<td></td>
</tr>
</tbody>
</table>

**Step 11**

<table>
<thead>
<tr>
<th>crypto ca authenticate name</th>
<th>Takes the name of the CA as the argument and authenticates it.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>• The following command output displays:</td>
</tr>
</tbody>
</table>
| Router(ca-trustpoint)# crypto ca authenticate mytp | Enter the base 64 encoded CA certificate.  
End with a blank line or the word "quit" on a line by itself. |
**Step 12**

Copy the following block of text containing the base 64 encoded CA certificate and paste it at the prompt.

```
MIIDIDCCAomgAwIBAgIENd70zzANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJw
UzEIAgIBAgIENd70zzANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJwUzEIAgIB
AgIENd70zzANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJwUzEIAgIBAgIENd70
A0zzANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJwUzEIAgIBAgIENd70zzANB
```

The following command output displays:

```
Certificate has the following attributes:

Fingerprint MD5: 67CB9DC0 13248A82 9BB2171E D11BECD4

Fingerprint SHA1: 67CB9DC0 13248A82 9BB2171E D11BECD4
```

**Step 13**

Enter **yes** to accept this certificate.

```
% Do you accept this certificate? [yes/no]: yes
```

The following command output displays:

```
Trustpoint CA certificate accepted.
```

```
% Certificate successfully imported
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 14</strong></td>
<td></td>
</tr>
<tr>
<td>serial-number</td>
<td>Specifies the router serial number in the certificate request.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>hostname1(ca-trustpoint)#</td>
<td></td>
</tr>
<tr>
<td>serial-number</td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td></td>
</tr>
<tr>
<td>revocation-check none</td>
<td>Specifies that certificate checking is ignored.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>hostname1(ca-trustpoint)#</td>
<td></td>
</tr>
<tr>
<td>revocation-check none</td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td></td>
</tr>
<tr>
<td>end</td>
<td>Exits ca-trustpoint configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>hostname1(ca-trustpoint)#</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 17</strong></td>
<td></td>
</tr>
<tr>
<td>trm register</td>
<td>Manually starts the Trend Micro Server registration process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>hostname1# trm register</td>
<td></td>
</tr>
</tbody>
</table>

**Configuring a Persistent Self-Signed Certificate for Enrollment via SSL**

This section contains the following tasks:

- **Note**
  These tasks are optional because if you enable the HTTPS server, it generates a self-signed certificate automatically using default values.

**Persistent Self-Signed Certificates Overview**

The SSL protocol can be used to establish a secure connection between an HTTPS server and a client (web browser). During the SSL handshake, the client expects the SSL server’s certificate to be verifiable using a certificate the client already possesses.

If Cisco IOS software does not have a certificate that the HTTPS server can use, the server generates a self-signed certificate by calling a PKI application programming interface (API). When the client receives...
this self-signed certificate and is unable to verify it, intervention is needed. The client asks you if the certificate should be accepted and saved for future use. If you accept the certificate, the SSL handshake continues.

Future SSL handshakes between the same client and the server use the same certificate. However, if the router is reloaded, the self-signed certificate is lost. The HTTPS server must then create a new self-signed certificate. This new self-signed certificate does not match the previous certificate, so you are once again asked to accept it.

Requesting acceptance of the router’s certificate each time that the router reloads may present an opportunity for an attacker to substitute an unauthorized certificate when you are being asked to accept the certificate. Persistent self-signed certificates overcome all these limitations by saving a certificate in the router’s startup configuration.

**Restrictions**

You can configure only one trustpoint for a persistent self-signed certificate.

**Note**

Do not change the IP domain name or the hostname of the router after creating the self-signed certificate. Changing either name triggers the regeneration of the self-signed certificate and overrides the configured trustpoint. WebVPN ties the SSL trustpoint name to the WebVPN gateway configuration. If a new self-signed certificate is triggered, then the new trustpoint name does not match the WebVPN configuration, causing the WebVPN connections to fail.

**Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters**

**Note**

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Perform the following task to configure a trustpoint and specify self-signed certificate parameters.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `crypto pki trustpoint` name
4. `enrollment selfsigned`
5. `subject-name [x.500-name]`
6. `rsakeypair` `key-label` `[key-size [encryption-key-size]]`
7. `crypto pki enroll` name
8. `end`
9. `show crypto pki certificates` `[trustpoint-name[verbose]]`
10. `show crypto pki trustpoints` `[status | label [status]]`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>enable</td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td>Example: Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Declares the CA that your router should use and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td>crypto pki trustpoint name</td>
<td>Note: Effective with Cisco IOS Release 12.3(8)T, the \texttt{crypto pki trustpoint} command replaced the \texttt{crypto ca trustpoint} command.</td>
</tr>
<tr>
<td>Example: Router(config)# crypto pki trustpoint local</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies self-signed enrollment.</td>
</tr>
<tr>
<td>enrollment selfsigned</td>
<td></td>
</tr>
<tr>
<td>Example: Router(ca-trustpoint)# enrollment selfsigned</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>(Optional) Specifies the requested subject name to be used in the certificate request.</td>
</tr>
<tr>
<td>subject-name [x.500-name]</td>
<td>- If no value for the \texttt{x-500-name} argument is specified, the FQDN, which is the default subject name, is used.</td>
</tr>
<tr>
<td>Example: Router(ca-trustpoint)# subject-name</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) Specifies which key pair to associate with the certificate.</td>
</tr>
<tr>
<td>rsakeypair key-label [key-size [encryption-key-size]]</td>
<td>- The value for the \texttt{key-label} argument will be generated during enrollment if it does not already exist or if the \texttt{auto-enroll regenerate} command was issued.</td>
</tr>
<tr>
<td>Example: Router(ca-trustpoint)# rsakeypair examplekey 2048</td>
<td>- Specify a value for the \texttt{key-size} argument for generating the key, and specify a value for the \texttt{encryption-key-size} argument to request separate encryption, signature keys, and certificates. The key-size and encryption-key-size must be the same size. Length of less than 2048 is no recommended.</td>
</tr>
<tr>
<td>Note</td>
<td>If this command is not enabled, the FQDN key pair is used.</td>
</tr>
</tbody>
</table>
### Configuring Certificate Enrollment for a PKI

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong> crypto pki enroll name</td>
<td>Tells the router to generate the persistent self-signed certificate.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(ca-trustpoint)# crypto pki enroll local</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> end</td>
<td>(Optional) Exits ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(ca-trustpoint)# end</td>
<td>• Enter this command a second time to exit global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 9</strong> show crypto pki certificates [trustpoint-name[verbose]]</td>
<td>Displays information about your certificate, the certification authority certificate, and any registration authority certificates.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show crypto pki certificates local verbose</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> show crypto pki trustpoints [status</td>
<td>label</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show crypto pki trustpoints status</td>
<td></td>
</tr>
</tbody>
</table>

### Enabling the HTTPS Server

Perform the following task to enable the HTTPS server.

**Before You Begin**

To specify parameters, you must create a trustpoint and configure it. To use default values, delete any existing self-signed trustpoints. Deleting all self-signed trustpoints causes the HTTPS server to generate a persistent self-signed certificate using default values as soon as the server is enabled.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. ip http secure-server
4. end
5. copy system:running-config nvram: startup-config

---

**Public Key Infrastructure Configuration Guide, Cisco IOS Release 15S**

118
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>terminal</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip http</td>
<td>Enables the HTTPS web server.</td>
</tr>
<tr>
<td>secure-server</td>
<td>Note: A key pair (modulus 1024) and a self-signed certificate are</td>
</tr>
<tr>
<td></td>
<td>automatically generated.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# ip http</td>
<td></td>
</tr>
<tr>
<td>secure-server</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> end</td>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> copy</td>
<td>Saves the self-signed certificate and the HTTPS server in enabled mode.</td>
</tr>
<tr>
<td>system:running-config</td>
<td></td>
</tr>
<tr>
<td>nvram: startup-config</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# copy system:</td>
<td></td>
</tr>
<tr>
<td>running-config nvram:</td>
<td></td>
</tr>
<tr>
<td>startup-config</td>
<td></td>
</tr>
</tbody>
</table>

Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment

Perform this task to configure a certificate enrollment profile for enrollment or reenrollment. This task helps you to configure an enrollment profile for certificate enrollment or reenrollment of a router with a Cisco IOS CA that is already enrolled with a third-party vendor CA.

Enable a router that is enrolled with a third-party vendor CA to use its existing certificate to enroll with the Cisco IOS certificate server so the enrollment request is automatically granted. To enable this functionality, you must issue the enrollment credential command. Also, you cannot configure manual certificate enrollment.

Before You Begin

Perform the following tasks at the client router before configuring a certificate enrollment profile for the client router that is already enrolled with a third-party vendor CA so that the router can reenroll with a Cisco IOS certificate server:
Defined a trustpoint that points to the third-party vendor CA.

Authenticated and enrolled the client router with the third-party vendor CA.

Note

- To use certificate profiles, your network must have an HTTP interface to the CA.
- If an enrollment profile is specified, an enrollment URL may not be specified in the trustpoint configuration. Although both commands are supported, only one command can be used at a time in a trustpoint.
- Because there is no standard for the HTTP commands used by various CAs, the user is required to enter the command that is appropriate to the CA that is being used.

> 

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. enrollment profile label
5. exit
6. crypto pki profile enrollment label
7. Do one of the following:
   • authentication url url
   • authentication terminal
8. authentication command
9. Do one of the following:
   • enrollment url url
   •
   • enrollment terminal
10. enrollment credential label
11. enrollment command
12. parameter number {value value | prompt string}
13. exit
14. show crypto pki certificates
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>crypto pki trustpoint name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# crypto pki trustpoint Entrust</td>
</tr>
<tr>
<td></td>
<td>Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>enrollment profile label</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(ca-trustpoint)# enrollment profile E</td>
</tr>
<tr>
<td></td>
<td>Specifies that an enrollment profile is to be used for certificate authentication and enrollment.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(ca-trustpoint)# exit</td>
</tr>
<tr>
<td></td>
<td>Exits ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>crypto pki profile enrollment label</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# crypto pki profile enrollment E</td>
</tr>
<tr>
<td></td>
<td>Defines an enrollment profile and enters ca-profile-enroll configuration mode.</td>
</tr>
<tr>
<td></td>
<td>• label --Name for the enrollment profile; the enrollment profile name must match the name specified in the enrollment profile command.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Do one of the following:</td>
</tr>
<tr>
<td></td>
<td>• authentication url url</td>
</tr>
<tr>
<td></td>
<td>• authentication terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(ca-profile-enroll)# authentication url <a href="http://entrust:81">http://entrust:81</a></td>
</tr>
<tr>
<td></td>
<td>Specifies the URL of the CA server to which to send certificate authentication requests.</td>
</tr>
<tr>
<td></td>
<td>• url --URL of the CA server to which your router should send authentication requests. If you are using HTTP, the URL should read &quot;http://CA_name,&quot; where CA_name is the host DNS name or IP address of the CA. If you are using TFTP, the URL should read &quot;tftp://certserver/file_specification.&quot; (If the URL does not include a file specification, the FQDN of the router will be used.)</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(ca-profile-enroll)# authentication terminal</td>
<td>Specifies manual cut-and-paste certificate authentication.</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>authentication command</strong></td>
</tr>
<tr>
<td><strong>Example:</strong> Router(ca-profile-enroll)# authentication command</td>
<td>(Optional) Specifies the HTTP command that is sent to the CA for authentication.</td>
</tr>
</tbody>
</table>
| **Step 9** | Do one of the following: 
| • enrollment url *url* | Specifies the URL of the CA server to which to send certificate enrollment requests via HTTP or TFTP. |
| • enrollment terminal | Specifies manual cut-and-paste certificate enrollment. |
| **Example:** Router(ca-profile-enroll)# enrollment url http://entrust:81/cda-cgi/clientcgi.exe | |
| **Example:** | |
| **Example:** Router(ca-profile-enroll)# enrollment terminal | |
| **Step 10** | enrollment credential *label* |
| **Example:** Router(ca-profile-enroll)# enrollment credential Entrust | (Optional) Specifies the third-party vendor CA trustpoint that is to be enrolled with the Cisco IOS CA. |
| **Note** | This command cannot be issued if manual certificate enrollment is being used. |
| **Step 11** | enrollment command |
| **Example:** Router(ca-profile-enroll)# enrollment command | (Optional) Specifies the HTTP command that is sent to the CA for enrollment. |
| **Step 12** | parameter *number* {value *value* | prompt *string*} | (Optional) Specifies parameters for an enrollment profile. |
### Purpose

**Command or Action**

*Example:*

```
Router(ca-profile-enroll)# parameter 1 value aaaa-bbbb-cccc
```

- This command can be used multiple times to specify multiple values.

### Step 13

**exit**

*Example:*

```
Router(ca-profile-enroll)# exit
```

- (Optional) Exits ca-profile-enroll configuration mode.

### Step 14

**show crypto pki certificates**

*Example:*

```
Router# show crypto pki certificates
```

- (Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.

---

**What to Do Next**

If you configured the router to reenroll with a Cisco IOS CA, you should configure the Cisco IOS certificate server to accept enrollment requests only from clients already enrolled with the specified third-party vendor CA trustpoint to take advantage of this functionality. For more information, see the module “Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment.”

**Configuration Examples for PKI Certificate Enrollment Requests**

**Configuring Certificate Enrollment or Autoenrollment Example**

The following example shows the configuration for the "mytp-A" certificate server and its associated trustpoint, where RSA keys generated by the initial autoenrollment for the trustpoint will be stored on a USB token, "usbtoken0":

```
crypto pki server mytp-A
database level complete
  issuer-name CN=company, L=city, C=country
  grant auto
! Specifies that certificate requests will be granted automatically.
! crypto pki trustpoint mytp-A
  revocation-check none
  rsakeypair myTP-A
  storage usbtoken0:
! Specifies that keys will be stored on usbtoken0:
  on usbtoken0:
! Specifies that keys generated on initial auto enroll will be generated on and stored on ! usbtoken0:
```

---

Public Key Infrastructure Configuration Guide, Cisco IOS Release 15S 123
### Configuring Autoenrollment Example

The following example shows how to configure the router to automatically enroll with a CA on startup, enabling automatic rollover, and how to specify all necessary enrollment information in the configuration:

```plaintext
crypto pki trustpoint trustpt1
enrollment url http://trustpt1.example.com/
subject-name OU=Spiral Dept., O=example.com
ip-address ethernet-0
serial-number none
usage ike
auto-enroll regenerate
password password1
rsa-key trustpt1 2048
!
crypto pki certificate chain trustpt1
certificate pki 0B
30820293 3082023D A0030201 0202010B 300D0609 2A864886 F700D101 04050030
79310B30 09060350 04061302 09196521 30130603 55040813 02430113 15301306
0355040A 130C4369 7363F20 05379734 656D3120 301E0603 55040813 17737562
6F762420 7462068 6168756C 57692049 05049535 31243022 06035504 31311B79
300D0609 2A864886 F700D101 04050030
00603550 0A130543 6973636F 3102010B 300D0609 2A864886 F700D101 04050030
!
quit
```

In this example, keys are neither regenerated nor rolled over.

---

### Configuring Certificate Autoenrollment with Key Regeneration Example

The following example shows how to configure the router to automatically enroll with the CA named "trustme1" on startup and enable automatic rollover. The `regenerate` keyword is issued, so a new key will be generated for the certificate and reissued when the automatic rollover process is initiated. The renewal percentage is configured as 90 so if the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires. The changes made to the running configuration are saved to the NVRAM startup configuration because autoenrollment will not update NVRAM if the running configuration has been modified but not written to NVRAM.

```plaintext
crypto pki trustpoint trustme1
enrollment url http://trustme1.example.com/
subject-name OU=Spiral Dept., O=example.com
ip-address ethernet0
serial-number none
auto-enroll 90 regenerate
password password1
```

---

**Note:** In this example, keys are neither regenerated nor rolled over.
rsakeypair trustme1 2048
exit
crypto pki authenticate trustme

copy system:running-config nvram:startup-config

Configuring Cut-and-Paste Certificate Enrollment Example

The following example shows how to configure certificate enrollment using the manual cut-and-paste enrollment method:

Router(config)#
crypto pki trustpoint TP
Router(ca-trustpoint)#
enrollment terminal
Router(ca-trustpoint)#
crypto pki authenticate TP

Enter the base 64 encoded CA certificate. End with a blank line or the word "quit" on a line by itself

BEGIN CERTIFICATE-----
MIICNDCCdgAwIBAgIQAQoQbq93hkiG9wOBAQFAQFADASMQswCQYDVQQGEwJQYzAAMBgNVBAoTC2NhcmVjdGlvblZ1dGVudDEzMSA8MB0GA1UEAxMjYmFpc3NjdhegZzAkGA1UdEwEB/wQQJMAwGA1UdIwQxMDUgZzAbBggrBgEFBQcDBQoGA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEB/BAIFVGZjMjFhMTBoMA0GCSqGSIb3DQEBCwUA...247

Certificate has the following attributes:
Fingerprint: D6C12961 CD78808A 4E02193C 0790082A

% Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.
% Certificate successfully imported

Router(config)#
crypto pki enroll TP
% Start certificate enrollment..
% The subject name in the certificate will be:
Router.example.com
% Include the router serial number in the subject name? [yes/no]: n
% Include an IP address in the subject name? [no]: n
Display Certificate Request to terminal? [yes/no]: y

Signature key request - Certificate Request follows:
MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTY5W5kQmFn2ZVyLmNpc2NVLmNv
btCbnzANBgkqhkiG9w0BAQEFAAOBjQAwjKgYEAEAxhXFD1Wan/h1Zs9z0tSzaKA
daoYUuoms9FE/Pew0ldh14VxXqgacst0s2Prx5k65PLPwpxvOiTPwYQM6ipLmyLwv

cjhyLTkYvohr6Dngcvk+G/5ohss9o9BkvONx042PqchFnx9EKMuZC7evwRkJEqR
mbHkXZBz8g3FjYqSJSWMCwEAAAaHMB8GSCgs6B3QD4EJJDJESMBAwDyVR0PFAQ/BAQDAgeAAMoQgCgSQibD3QEBBAUAAGBMAM6tYFw95F0Y7U7C4YH1vRUfF4SC
hR1A8rj1jD6Po391f7ryFUIP3tBnIZS55BHUt3Mfg89sgodaWubmPdBzahC1q0D6
0B7fJnLwnIDStOv5jKogPHk1isEGzMBosUW9lJienQNdEPhJc5lTDcDvciA6j0
N18rOktNmt8Q+

% Display enrollment request? [yes/no]: y
Encryption key request - Certificate Request follows:
MIIBhTCB7wIBADAlMSMwIQYJKoZIhvcNAQkCFhRTY5W5kQmFn2ZVyLmNpc2NVLmNv
btCbnzANBgkqhkiG9w0BAQEFAAOBjQAwjKgYEAEAxhXFD1Wan/h1Zs9z0tSzaKA
daoYUuoms9FE/Pew0ldh14VxXqgacst0s2Prx5k65PLPwpxvOiTPwYQM6ipLmyLwv

Public Key Infrastructure Configuration Guide, Cisco IOS Release 15S

125
Configuring Certificate Enrollment for a PKI

Configuring Cut-and-Paste Certificate Enrollment Example

You can verify that the certificate was successfully imported by issuing the show crypto pki certificates command:

Router# show crypto pki certificates
Certificate
Status: Available
Certificate Serial Number: 14DECE050000000001
Certificate Usage: Encryption
Issuer:
CN = TPCA-root
O = Company
Public Key Infrastructure Configuration Guide, Cisco IOS Release 15S

126
Configuring Manual Certificate Enrollment with Key Regeneration Example

The following example shows how to regenerate new keys with a manual certificate enrollment from the CA named "trustme2":

crypto pki trustpoint trustme2
enrollment url http://trustme2.example.com/
subject-name OU=Spiral Dept., O=example.com
ip-address ethernet0
serial-number none
regenerate
password password1
rsakeypair trustme2 2048
exit
crypto pki authenticate trustme2
crypto pki enroll trustme2
Creating and Verifying a Persistent Self-Signed Certificate Example

The following example shows how to declare and enroll a trustpoint named “local” and generate a self-signed certificate with an IP address:

crypto pki trustpoint local
  enrollment selfsigned
end
configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
crypto pki enroll local
Nov 29 20:51:13.067: %SSH-5-ENABLED: SSH 1.99 has been enabled
Nov 29 20:51:13.267: %CRYPTO-6-AUTOGEN: Generated new 512 bit key pair
% Include the router serial number in the subject name? [yes/no]: yes
% Include an IP address in the subject name? [no]: yes
Enter Interface name or IP Address[]: ethernet 0
Generate Self Signed Router Certificate? [yes/no]: yes
Router Self Signed Certificate successfully created

Note: A router can have only one self-signed certificate. If you attempt to enroll a trustpoint configured for a self-signed certificate and one already exists, you receive a notification and are asked if you want to replace it. If so, a new self-signed certificate is generated to replace the existing one.

Enabling the HTTPS Server Example

The following example shows how to enable the HTTPS server and generate a default trustpoint because one was not previously configured:

configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
ip http secure-server
% Generating 1024 bit RSA keys ...[OK]
*Dec 21 19:14:15.421:%PKI-4-NOAUTOSAVE:Configuration was modified. Issue "write memory"
to save new certificate
Router(config)#

Note: You need to save the configuration to NVRAM if you want to keep the self-signed certificate and have the HTTPS server enabled following router reloads.

The following message also appears:

*Dec 21 19:14:10.441:%SSH-5-ENABLED:SSH 1.99 has been enabled

Note: Creation of the key pair used with the self-signed certificate causes the Secure Shell (SSH) server to start. This behavior cannot be suppressed. You may want to modify your Access Control Lists (ACLs) to permit or deny SSH access to the router. You can use the ip ssh rsa keypair-name unexisting-key-pair-name command to disable the SSH server.
Verifying the Self-Signed Certificate Configuration Example

The following example displays information about the self-signed certificate that you just created:

Router# `show crypto pki certificates`
Router Self-Signed Certificate
Status: Available
Certificate Serial Number: 01
Certificate Usage: General Purpose
Issuer:
   cn=IOS-Self-Signed-Certificate-3326000105
Subject:
   Name: IOS-Self-Signed-Certificate-3326000105
   cn=IOS-Self-Signed-Certificate-3326000105
Validity Date:
   start date: 19:14:14 GMT Dec 21 2004
   end date: 00:00:00 GMT Jan 1 2020
Associated Trustpoints: TP-self-signed-3326000105

Note
The number 3326000105 is the router’s serial number and varies depending on the router’s actual serial number.

The following example displays information about the key pair corresponding to the self-signed certificate:

Router# `show crypto key mypubkey rsa`
% Key pair was generated at: 19:14:10 GMT Dec 21 2004
Key name: TP-self-signed-3326000105
Usage: General Purpose Key
Key is not exportable.
Key Data:
   30819F30 0D06092A 864886F7 0D010101 05000381 89028181 00B88F70
   6BC7B6ED 7606CFF3 135C1D91 8F360292 CA44A032 5AC1A8FD 095E4865 F8C95A2B
   BFDDC8B7 E64A3804 00BB7326 20BD456 1BA8B78B 00D2560C B092E9AE
   06C8F8B0 66727F3A 5CD8E892 9D4C4F71 38BC8269 214F6299 4BA8FABF 9486B7FF
   2B041BCA 550999AF 2FEE12A5 67B669A 2D88AB77 3988DE0E AA258C8C B7020501 0001
% Key pair was generated at: 19:14:13 GMT Dec 21 2004
Key name: TP-self-signed-3326000105.server
Usage: Encryption Key
Key is not exportable.
Key Data:
   307C300D 0D06092A 4886F70D 0D010101 05000381 00C56808 9777B842
   463E5783 FE96E49E F446D787 7F099AF3 EA266651 56EE29F4 5B030D93 2FC9F81D
   8A46621F 3FAC2F3 046ED9DD C5F27C20 1BA6B3B9 08F16E45 C33D3D37 F863D605
   34E3040E B4921BC5 DMC9EBBA 50C54AA0 BF551BDD B8453F50 81020001 0001

Note
The second key pair with the name TP-self-signed-3326000105.server is the SSH key pair and is generated when any key pair is created on the router and SSH starts up.

The following example displays information about the trustpoint named "local":

Router# `show crypto pki trustpoints`
Trustpoint local:
   Subject Name:
      serialNumber=C63EBBE9+ipaddress=10.3.0.18+hostname=test.example.com
   Serial Number: 01
   Persistent self-signed certificate trust point
Configuring Direct HTTP Enrollment Example

The following example show how to configure an enrollment profile for direct HTTP enrollment with a CA server:

crypto pki trustpoint Entrust
enrollment profile E
serial
crypto pki profile enrollment E
authentication url http://entrust:81
authentication command GET /certs/cacert.der
enrollment url http://entrust:81/cda-cgi/clientcgi.exe
enrollment command POST reference_number=$P2&authcode=$P1
&retrievedAs=rawDER&action=getServerCert&pkcs10Request=$REQ
parameter 1 value aaaa-bbbb-cccc
parameter 2 value 5001

Additional References

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB token RSA operations: Benefits of using USB tokens</td>
<td>“Storing PKI Credentials” module in the Cisco IOS Security Configuration Guide: Secure Connectivity</td>
</tr>
<tr>
<td>Overview of PKI, including RSA keys, certificate enrollment, and CAs</td>
<td>“Cisco IOS PKI Overview: Understanding and Planning a PKI” module in the Cisco IOS Security Configuration Guide: Secure Connectivity</td>
</tr>
<tr>
<td>Secure Device Provisioning: functionality overview and configuration tasks</td>
<td>“Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI” module in the Cisco IOS Security Configuration Guide: Secure Connectivity</td>
</tr>
<tr>
<td>RSA key generation and deployment</td>
<td>“Deploying RSA Keys Within a PKI” module in the Cisco IOS Security Configuration Guide: Secure Connectivity</td>
</tr>
<tr>
<td>Related Topic</td>
<td>Document Title</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Setting up and using a USB token</td>
<td>“Storing PKI Credentials” module in the Cisco IOS Security Configuration Guide: Secure Connectivity</td>
</tr>
<tr>
<td>Cisco IOS security commands</td>
<td>Cisco IOS Security Command Reference</td>
</tr>
<tr>
<td>Suite-B ESP transforms</td>
<td>Configuring Security for VPNs with IPsec feature module.</td>
</tr>
<tr>
<td>Suite-B SHA-2 family (HMAC variant) and Elliptic Curve (EC) key pair configuration.</td>
<td>Configuring Internet Key Exchange for IPsec VPNs feature module.</td>
</tr>
<tr>
<td>Suite-B Integrity algorithm type transform configuration.</td>
<td>Configuring Internet Key Exchange Version 2 (IKEv2) feature module.</td>
</tr>
<tr>
<td>Suite-B Elliptic Curve Digital Signature Algorithm (ECDSA) signature (ECDSA-sig) authentication method configuration for IKEv2.</td>
<td>Configuring Internet Key Exchange Version 2 (IKEv2) feature module.</td>
</tr>
<tr>
<td>Suite-B Elliptic curve Diffie-Hellman (ECDH) support for IPsec SA negotiation</td>
<td>Configuring Internet Key Exchange for IPsec VPNs and Configuring Internet Key Exchange Version 2 (IKEv2) feature modules.</td>
</tr>
<tr>
<td>Recommended cryptographic algorithms</td>
<td>Next Generation Encryption</td>
</tr>
</tbody>
</table>

**MIBs**

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None.</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

**Technical Assistance**

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>
Feature Information for PKI Certificate Enrollment

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4: Feature Information for PKI Certificate Enrollment

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS USB Token PKI Enhancements--Phase 2</td>
<td>12.4(11)T</td>
<td>This feature enhances USB token functionality by using the USB token as a cryptographic device. USB tokens may be used for RSA operations such as key generation, signing, and authentication. The following section provides information about this feature:</td>
</tr>
</tbody>
</table>

- Configuring Certificate Enrollment or Autoenrollment, on page 98

Note: This document describes the use of utilizing USB tokens for RSA operations during initial autoenrollment for a trustpoint. For other documents on this topic, see the "Feature Information for PKI Certificate Enrollment, on page 132" section.
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Certificate Authority Key Rollover | 12.4(2)T | This feature introduces the ability for root CAs to roll over expiring CA certificates and keys and to have these changes propagate through the PKI network without manual intervention. The following sections provide information about this feature:  
  - Automatic Certificate Enrollment, on page 97  
  - Configuring Certificate Enrollment or Autoenrollment, on page 98  
  The following commands were introduced or modified by this feature: auto-rollover, crypto pki certificate chain, crypto pki export pem, crypto pki server, crypto pki server info request, show crypto pki certificates, show crypto pki server, show crypto pki trustpoint. |
| Certificate Autoenrollment         | 12.2(8)T | This feature introduces certificate autoenrollment, which allows the router to automatically request a certificate from the CA that is using the parameters in the configuration. The following sections provide information about this feature:  
  - Automatic Certificate Enrollment, on page 97  
  - Configuring Certificate Enrollment or Autoenrollment, on page 98  
  The following commands were introduced by this feature: auto-enroll, rsakeypair, show crypto ca timers. |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Certificate Enrollment Enhancements  | 12.2(8)T  | This feature introduces five new `crypto ca trustpoint` commands that provide new options for certificate requests and allow users to specify fields in the configuration instead of having to go through prompts. The following section provides information about this feature:  
   - Configuring Certificate Enrollment or Autoenrollment, on page 98  
   
   The following commands were introduced by this feature: `ip-address(ca-trustpoint), password(ca-trustpoint), serial-number, subject-name, usage`. |
| Direct HTTP Enrollment with CA Servers | 12.3(4)T  | This feature allows users to configure an enrollment profile if their CA server does not support SCEP and they do not want to use an RA-mode CS. The enrollment profile allows users to send HTTP requests directly to the CA server instead of to an RA-mode CS. The following sections provide information about this feature:  
   - Certificate Enrollment Profiles, on page 98  
   - Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment, on page 119  
   
   The following commands were introduced by this feature: `authentication command, authentication terminal, authentication url, crypto ca profile enrollment, enrollment command, enrollment profile, enrollment terminal, enrollment url, parameter`. |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Import of RSA Key Pair and Certificates in PEM Format | 12.3(4)T | This feature allows customers to issue certificate requests and receive issued certificates in PEM-formatted files.  
The following section provides information about this feature:  
  - Configuring Manual Certificate Enrollment, on page 104  
  The following commands were modified by this feature: enrollment, enrollment terminal. |
| Key Rollover for Certificate Renewal              | 12.3(7)T | This feature allows the certificate renewal request to be made before the certificate expires and retains the old key and certificate until the new certificate is available.  
The following sections provide information about this feature:  
  - Automatic Certificate Enrollment, on page 97  
  - Configuring Certificate Enrollment or Autoenrollment, on page 98  
  - Configuring Manual Certificate Enrollment, on page 104  
  The following commands were introduced or modified by this feature: auto-enroll, regenerate. |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual Certificate Enrollment (TFTP Cut-and-Paste)</td>
<td>12.2(13)T</td>
<td>This feature allows users to generate a certificate request and accept CA certificates and the router’s certificates via a TFTP server or manual cut-and-paste operations.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following sections provide information about this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Supported Certificate Enrollment Methods, on page 95</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Configuring Manual Certificate Enrollment, on page 104</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following commands were introduced or modified by this feature: crypto ca import, enrollment, enrollment terminal.</td>
</tr>
<tr>
<td>Multiple-Tier CA Hierarchy</td>
<td>12.2(15)T</td>
<td>This enhancement enables users to set up a PKI in a hierarchical framework to support multiple CAs. Within a hierarchical PKI, all enrolled peers can validate the certificate of one another as long as the peers share a trusted root CA certificate or a common subordinate CA.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following section provides information about this enhancement:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Framework for Multiple CAs, on page 94</td>
</tr>
<tr>
<td>Note</td>
<td></td>
<td>This is a minor enhancement. Minor enhancements are not typically listed in Feature Navigator.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Persistent Self-Signed Certificates | 12.2(33)SXH 12.2(33)SRA 12.3(14)T | This feature allows the HTTPS server to generate and save a self-signed certificate in the router startup configuration. Thus, future SSL handshakes between the client and the HTTPS server can use the same self-signed certificate without user intervention. The following sections provide information about this feature:  
  - Supported Certificate Enrollment Methods, on page 95  
  - Configuring a Persistent Self-Signed Certificate for Enrollment via SSL, on page 115  
  The following commands were introduced or modified by this feature:  
    - enrollment selfsigned, show crypto pki certificates, show crypto pki trustpoints. |
| PKI Status                   | 12.3(11)T    | This enhancement adds the status keyword to the show crypto pki trustpoints command, which allows you to display the current status of the trustpoint. Prior to this enhancement, you had to issue the show crypto pki certificates and the show crypto pki timers commands for the current status. The following section provides information about this enhancement:  
  - How to Configure Certificate Enrollment for a PKI, on page 98 |
<p>| Note                        |              | This is a minor enhancement. Minor enhancements are not typically listed in Feature Navigator.                                                      |</p>
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Reenroll Using Existing Certificates | 12.3(11)T| This feature allows users to reenroll a router with a Cisco IOS CA via existing certificates from a third-party vendor CA. The following section provides information about this enhancement:  
  • Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment, on page 119  
  The following commands were introduced by this feature: enrollment credential, grant auto trustpoint. |
<p>| Trustpoint CLI                       | 12.2(8)T | This feature introduces the crypto pki trustpoint command, which adds support for trustpoint CAs.                                                      |</p>
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Suite-B support in IOS SW crypto | 15.1(2)T | Suite-B adds the following support for certificate enrollment for a PKI:  
- Elliptic Curve Digital Signature Algorithm (ECDSA) (256 bit and 384 bit curves) is used for the signature operation within X.509 certificates.  
- PKI support for validation of for X.509 certificates using ECDSA signatures.  
- PKI support for generating certificate requests using ECDSA signatures and for importing the issued certificates into IOS.  
Suite-B requirements comprise of four user interface suites of cryptographic algorithms for use with IKE and IPSec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm. See the Configuring Security for VPNs with IPSec feature module for more detailed information about Cisco IOS Suite-B support. |
| Public Key Infrastructure (PKI) IPv6 Support for VPN Solutions | 15.2(1)T | The enrollment url (ca-trustpoint) command was modified to allow the specification of an IPv6 address in the URL for the CA.  
The ip-address (ca-trustpoint) command was modified to allow the specification of an IPv6 address that is included as "unstructuredAddress" in the certificate request. }
trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)
CHAPTER 5

Setting Up Secure Device Provisioning for Enrollment in a PKI

This module describes how to use Secure Device Provisioning (SDP) in a public key infrastructure (PKI). SDP is a web-based certificate enrollment interface that can be used to easily deploy PKI between two end devices, such as a Cisco IOS client and a Cisco IOS certificate server. The end devices may or may not be directly connected to the network at the time of deployment or provisioning. SDP provides a solution for users deploying a large number of peer devices (including certificates and configurations).

Note
Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

- Finding Feature Information, page 141
- Prerequisites for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI, page 142
- Information About Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI, page 143
- How to Set Up Secure Device Provisioning (SDP) for Enrollment in a PKI, page 169
- Configuration Examples for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI, page 187
- Additional References, page 196
- Feature Information for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI, page 197

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.
Prerequisites for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI

Setting Up SDP for Enrollment in a PKI
Before you set up SDP, your environment should meet the following requirements:

• The petitioner device and the server must have IP connectivity between each other.
• The introducer must have a web browser that supports JavaScript.
• The introducer must have enable privileges on the client device.
• A Cisco IOS Release 12.3(8)T PKI-enabled image or a later image.

Setting Up SDP for Enrollment in a PKI Using USB Tokens
To leverage USB tokens to provision devices with SDP, your environment should meet the following requirements:

• Both the petitioner device and the server must have IP connectivity between each other.
• The introducer must have a web browser that supports JavaScript.
• The introducer must have enable privileges on the client device.
• The introducer must have access to a petitioner device.
• The introducer must have access to the USB token and PIN, if configured.
• A Cisco IOS Release 12.4(15)T PKI-enabled image or a later image.

Note
Cisco IOS Release 12.4(15)T or a later release provides the flexibility to move credentials stored on the USB token. However, the device used to configure the USB token may run any Cisco IOS Release 12.3(14)T PKI-enabled image or a later image.

Using SDP to Configure a Device for an Internet Connection Through a Service Provider
To leverage SDP to configure a device that is not connected to the Internet, your environment should meet the following requirements:

• The introducer must have a web browser that supports JavaScript.
• The introducer must have enable privileges on the client device.
• A Cisco router that supports a DHCP client and a PPPoE client and has a configured LAN or WAN interface.
A Cisco IOS Release 12.4(20)T PKI-enabled image or a later image. If a previous Cisco IOS release is used on one of the devices, the SDP functionality defaults to the earlier Cisco IOS version.

Information About Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI

SDP Overview

SDP (also referred to as Trusted Transitive Introduction [TTI]) is a communication protocol that provides a bidirectional introduction between two end entities, such as a new network device and a Virtual Private Network (VPN). SDP involves the following three entities (see the figure below):

- **Introducer**—A mutually trusted device that introduces the petitioner to the registrar. The introducer can be a device user, such as a system administrator.
  - An introducer can be configured as an administrative introducer, which allows an administrator performing the introduction to supply the name for the device being introduced. The supplied device name is used as if it were the name of an introducer in the normal SDP mechanisms, preserving the existing functionality of the SDP configuration. For more information on function of the administrative introducer, see the section *Authentication and Authorization Lists for an Administrative Introducer*, on page 155.

- **Petitioner**—A client, or new device, to be introduced to the secure network.

- **Registrar**—A server that authorizes the petitioner. The registrar can be a certificate server.

*Figure 4: Post-Introduction Secure Communication*

As of Cisco IOS Release 12.4(20)T or a later release, the introducer can start the SDP process without establishing prior Internet connectivity on the petitioner. The use of the prep-connect phase and the connect phase provides the ability to configure a petitioner for Internet connectivity through a service provider. See the *How SDP Works*, on page 144 for more information on the prep-connect phase and the connect phase.

The registrar communicates directly with an external authentication, authorization, and accounting (AAA) server to verify petitioner credentials, permit or deny enrollment, and retrieve specific petitioner configuration...
information. The petitioner and registrar serve web pages to the introducer, the end user. The petitioner receives the bootstrap configuration from a remote management system through the introducer’s web browser.

SDP is implemented over a web browser with six possible phases—prep-connect (optional), connect, start (optional), welcome, introduction, and completion. Each phase is shown to the user through a web page. See the How SDP Works, on page 144 for more information on each phase.

How SDP Works

The following sections describe how SDP deploys PKI between two devices:

- **SDP Prep-Connect Phase, on page 144**
- **SDP Connect Phase, on page 146**
- **SDP Start Phase, on page 148**
- **SDP Welcome Phase, on page 149**
- **SDP Introduction Phase, on page 149**
- **SDP Completion Phase, on page 150**

The SDP process starts with one of three entry pages being loaded into the web browser by the introducer: the SDP prep-connect phase received from the administrator; the start phase loaded from the registrar; or the welcome phase loaded from the petitioner.

The sample figures show how to introduce the local device (the petitioner) to the secure domain of the registrar. The “introducer” is referred to as the end user.

SDP Prep-Connect Phase

The prep-connect page is optional. Without the prep-connect page, the petitioner must have IP connectivity established.

The administrator must configure the prep-connect template and send the prep-connect page to the introducer. See the Default Prep-Connect Template, on page 160 for more information.

The administrator must also obtain and communicate the username and password for the secure network to the introducer by a telephone call, an e-mail, a secure e-mail, a CD, or a USB token. The registrar may be configured to authenticate the introducer using an existing AAA infrastructure (for example, an existing username and password database that is part of the existing corporate domain). The SDP prep-connect phase supports a challenge password mechanism as is used by common AAA infrastructures. See the How SDP Uses an External AAA Database, on page 154 for more information.
After receiving the prep-connect page, the introducer must load the page onto the computer where the HTTP browser operates. The introducer then loads the prep-connect page into the HTTP browser as a local file and then the prep-connect page is displayed (see the figure below).

*Figure 5: Sample SDP Prep-Connect Page*

![Sample SDP Prep-Connect Page](image)

After the introducer clicks the Log onto Cisco Device button, the login dialog box is displayed (see the figure below). The introducer enters the factory default username (cisco) and password (cisco) of the Cisco device.

*Figure 6: Sample Petitioner Login Dialog Box*

![Sample Petitioner Login Dialog Box](image)

The introducer authenticates with the petitioner and then Internet connectivity is tested by attempting to access a known URL. Access to www.cisco.com (198.133.219.25) is tested by default. The administrator can modify the URL to be used for testing connectivity by modifying the default prep-connect template. For more information about modifying the default test URL and other fields that the administrator may configure for the prep-connect page, see the section Default Prep-Connect Template, on page 160.

**Note**

To mitigate the possibility that the prep-connect page could be modified to contain an IP address of an untrusted registrar or that a prep-connect page might be e-mailed from an untrusted source, use a secure method, such as secure e-mail, to send the prep-connect page.

If Internet connectivity is established either the start page or welcome page is displayed, depending on the prep-connect template setting as defined by the administrator. If Internet connectivity is not established, the connect page is displayed.
SDP Connect Phase

The connect page is displayed only if the prep-connect page is used and there is no IP connectivity for the petitioner at the completion of the prep-connect phase. The connect page has three IP address assignment methods to allow flexibility for your Cisco IOS platform: Dynamic Host Configuration Protocol (DHCP), Point to Point Protocol over Ethernet (PPPoE), or static IP address assignment.

Note:
SDP functionality is not used with the Cisco IOS configuration to establish Internet connectivity. SDP functionality includes a signature on the Cisco IOS configuration, guaranteeing that the values have not changed in transit.

DHCP IP Address Assignment Method

If the introducer chooses DHCP, the default method, for the IP address assignment method option (see the figure below), clicking the Connect button causes the petitioner to be configured for Internet connectivity.

Figure 7: Sample Connect Page for DHCP IP Address Assignment Method
PPPoE IP Address Assignment Method

If the introducer chooses PPPoE, input fields for PPPoE username and password are displayed (see the figure below). The introducer must enter the username and password as supplied by the Internet service provider (ISP) and then click the Connect button, which causes petitioner to be configured for Internet connectivity.

Figure 8: Sample Connect Page for PPPoE IP Address Assignment Method

Static IP Address Assignment Method

If the introducer chooses static, input fields for the IP address, netmask, and the default gateway are displayed (see the figure below). The introducer must enter the configuration values as supplied by the ISP and then click the Connect button, which causes petitioner to be configured for Internet connectivity.

Figure 9: Connect Page for Static IP Address Assignment Method

Connect Page IP Address Configuration

After IP address configuration, Internet connectivity is tested again by attempting to access a known URL configured by the administrator in the prep-connect template (www.cisco.com by default). If Internet connectivity is now established either the start page or welcome page is displayed, depending on the
prep-connect template setting as defined by the administrator. If Internet connectivity is not established, the introducer should verify the settings entered or contact their administrator.

**SDP Start Phase**

The start page is optional. Without the start page, during the SDP exchange, the user clicks the Next button on the welcome page and is sent to the registrar’s introduction page. Because the user has not previously connected to the registrar, he or she is required to log in to the registrar using available credentials (per the registrar configuration). Some browsers fail to reconnect to the registrar after the user has entered the login data. As of Cisco IOS Release 12.4(4)T, users may configure their browsers to begin the SDP exchange by contacting the registrar’s introduction URL through a start page. Thereafter, the registrar can direct the user to the welcome page, which is on the petitioner device. The SDP transaction continues through the welcome, introduction, and completion phases as described in this document.

To begin the SDP transaction from the registrar, the user must configure the browser through the `template http start` command; otherwise, the SDP transaction must begin from the welcome page on the petitioner. See the *How Custom Templates Work with SDP*, on page 156.

Before the welcome page is displayed, the user must direct his or her browser to the start page through the URL `http://registrar/ezsdd/intro`. A login dialog box is then displayed, and the end user can log into the registrar through a username and password supplied by the administrator to access the secure network (see the figure below).

*Figure 10: Registrar Remote Login Dialog Box*

![Registrar Remote Login Dialog Box](image)

After entering a valid username and password, the start page is displayed (see the figure below).

*Figure 11: Sample SDP Start Page*

![Sample SDP Start Page](image)
The user must log into the petitioner through the URL http://10.10.10.1/ezsdd/welcome. The welcome phase begins when the user clicks the Next button on the start page.

**SDP Welcome Phase**

The local login dialog box is then displayed (see the figure below), and the end user can log into the local device through the factory default username (cisco) and password (cisco). The welcome page is then displayed.

*Figure 12: Petitioner Local Login Dialog Box*

![Authentication Required](image)

After the password is successfully entered, the welcome web page is displayed (see the figure below), which is served by the petitioner.

*Figure 13: Sample SDP Welcome Page*

![Welcome to Cisco Secure Device Deployment](image)

After entering the URL of the registrar (for example, http://192.0.2.155/ezsdd/intro) and clicking the Next button on the welcome web page, the SDP introduction phase begins and the introduction page, which is served by the registrar, is displayed.

**SDP Introduction Phase**

Before the introduction page is displayed, the end user must log into the registrar if the user has not already done so from the start page (see “SDP Start Phase, on page 148”), which utilizes the external AAA database.

With an external AAA database, the introducer can use an account on the database to perform the introduction without requiring knowledge of the enable password of the registrar. Without an external AAA database, the introducer may use the enable password of the registrar for authentication.
Using the enable password of the registrar exposes the password to end users; therefore, it is recommended that the enable password be used for administrative testing only.

The administrative introducer is identified by the HTTP authentication for the introduction page (or the start page), with the AAA database query returning administrative privilege for the user. If the introducer has administrator privilege, the device name is that which was entered in the administrative introduction page. If the introducer does not have administrative privileges, the device name is the introducer name. The existing device certificate is the current certificate on the petitioner, which may be the manufacturing identification certificate (MIC). This certificate may or may not exist. For more information on the function of the external AAA database, see the section “How SDP Uses an External AAA Database, on page 154.”

After the end user successfully enters his or her password, the introduction web page is displayed (see the figure below).

**Figure 14: Sample SDP Introduction Page**

At this point, the registrar passes device information to the external management system to obtain a bootstrap configuration file. For more information on options available to identify a customized bootstrap configuration file, see the section **Custom HTML Template Expansion Rules, on page 157.**

After the end user clicks the Next button on the introduction page, the end user enters the completion phase and automatically returns to his or her local device.

**SDP Completion Phase**

Now that the end user has enrolled the petitioner with the registrar, the petitioner serves the completion page (see the figure below).

**Figure 15: Sample SDP Completion Page**
The SDP exchange is now complete. The petitioner has received configuration information from the registrar and should receive a certificate from the registrar shortly.

**SDP Leveraging USB Tokens**

SDP provides for highly scalable deployments and streamlines the deployment of an individual device or multiple devices. USB tokens provide for secure storage and configuration distribution.

As of Cisco IOS Release 12.4(15)T or a later release, USB tokens may be utilized to transfer PKI credentials using SDP to a remote device, and SDP may be used to configure the USB token. The USB token may then be used to provision a device at the same location, or the USB token may be transported to another location where it may be used to provision a remote device. For more information about configuring and using USB tokens, see the SDP Leveraging USB Tokens, on page 151 section.

An example SDP deployment using a USB token to transfer PKI credentials is shown in the figure below. The required devices include the USB token and the SDP entities required to provision a device. These SDP entities are the introducer, the registrar, a petitioner at the local location, Petitioner A, and a petitioner at the remote location, Petitioner B. Optionally, a management server may be used.
An optional configuration would be to configure one device as both the registrar and a petitioner, which may be beneficial when the USB token is transported to a remote location. The remote location would not require a separate petitioner device.

**Figure 16: Example SDP Environment Using USB Tokens to Transfer Credentials**

Use of SDP to Configure the USB Token

Prior to initiating an SDP introduction a USB token is inserted into the petitioner device. In the example configuration shown in Use of SDP to Configure the USB Token, the USB token would be inserted into Petitioner A. The petitioner may be configured to ignore any existing information on the USB token. As in regular SDP operations, for a scalable configuration of USB tokens, an initial template configuration has to be prepared and placed onto each SDP device with appropriate target configuration information.

Files used to provision a device are moved in the following sequence, shown by the numbered arrows in Use of SDP to Configure the USB Token.

1. One petitioner, Petitioner A, is at the local location. Petitioner A engages directly with the SDP exchange to perform the initial configuration of the USB token. Files used to configure the USB token, binary files and template files, are retrieved from the registrar and moved to Petitioner A.
The URL for the binary file location is expanded on the registrar. Binary files are not processed through the template expansion functions. The template expansion occurs on the registrar for both the source URL and destination URL.

By default, binary files and template files are retrieved from and stored to NVRAM on the registrar and petitioner respectively. The binary file location on the registrar and the destination binary file location on Petitioner A may be specified with the `binary file` command. The template file location on the registrar and the destination template file location on Petitioner A may be specified with the `template file` command.

1. The Rivest, Shamir, and Adelman (RSA) keys and certificate chain information are moved from Petitioner A to the USB token.
2. The USB token is transported to the remote location where it is inserted into Petitioner B.
3. The configuration files on the USB token are used to provision the local device. Files from the USB token may be moved to a storage location on Petitioner B with the `crypto key move rsa` command.

**SDP Phases with a USB Token**

The same SDP phase concepts introduced in the "SDP Overview" section are used, with the following distinctions in the SDP welcome phase, the SDP introduction phase, and the SDP completion phase.

**SDP Welcome Phase with a USB Token**

The SDP welcome phase begins as usual, when an introduction is initiated by connecting to the welcome user interface. If there is an existing certificate on the USB token, it is used for signing the SDP exchange. Instead of a local RSA key pair, a new RSA key pair on the token is used.

**Note**

The RSA key pair generation may take a substantial length of time, anywhere from 5 to 10 minutes if the key is generated on the token. The length of time is dependent on hardware key generation routines available on the USB token. An informative web page is presented to the introducer, indicating that RSA key pair generation is occurring.

The new key pair generated by Petitioner A is added to the USB token without removing any existing RSA key pairs. SDP AV pairs indicate both that a token is being used and if there is any token secondary configuration information. If an optional management server is in use, the AV pair information is used to determine if any special configuration commands are needed.

**SDP Introduction Phase with a USB Token**

The SDP Introduction phase begins with AV pairs being transferred to the registrar. When the registrar detects USB token related AV pairs, the registrar, if previously configured, may prepare configuration information destined for the USB token. Currently configuration commands are sent as a specific configuration files that are subsequently merged with the running configuration.

The administrator can leverage normal SDP configuration commands to configure the USB token. USB token information that should be configured includes the certificate, the bootstrap configuration, and the PIN number configuration.

**SDP Completion Phase with a USB Token**

At the beginning of the completion phase, the introduction proceeds with AV pairs being transferred to the petitioner (in *SDP Phases with a USB Token*, this would be Petitioner A). The various files are stored in the
specified file system locations and then the existing configuration file processing proceeds. This ordering allows the configuration to take advantage of the new files that have been transferred.

**Use of the Configured USB Token**

After the USB token is configured by Petitioner A, it is transported from its current location to the remote location, where the second petitioner, Petitioner B is located. The USB token is inserted into the target device, Petitioner B, which then inherits the USB token configuration and cryptographic material from the USB token. The end user at the remote location must have the PIN number on the USB token. The PIN number is either the default factory PIN or the PIN number the administrator configured during the introduction phase.

**How SDP Uses an External AAA Database**

The external AAA database is accessed twice during the SDP exchange. The first time the AAA database is accessed, the introducer is authenticated; that is, when the registrar receives an introduction request through the secure HTTP (HTTPS) server, the registrar does an AAA lookup based on the introducer’s username and password to authorize the request. The second time the AAA database is accessed, authorization information is obtained and applied to the configuration and certificates that are issued to the petitioner device; that is, the registrar checks the integrity of the request by verifying the request signature using the petitioner-signing certificate. The certificate subject name may be specified in the AAA database, and up to nine configuration template variables may be specified and expanded into the template configuration.

**Use of a Self-Signed Certificate Versus a Certificate Issued by Another CA Server**

By default, the SDP exchange results in only one certificate being issued to the petitioner device. Although just one certificate is issued, the introducer is not restricted from introducing multiple devices and thus obtaining multiple certificates. By specifying the subject name in the certificate that is issued, you can be assured that all certificates that are issued in this way are associated with the introducer. You can use PKI AAA integration to further restrict the use of these certificates. Additionally, the AAA database can be configured to accept only one authentication and authorization request per user.

Because the petitioner certificate is self-signed, it is just used to convey the public key of the petitioner. No verification or authorization check is performed on the certificate; thus, authorization is per-user based and no per-device information is used.

There are some scenarios when per-device authorization is preferred. Therefore, if the petitioner is able to use certificates issued by other certification authority (CA) servers for SDP transactions, the existing PKI can be used and authorization can be achieved over the certificate attributes.

Configuring the petitioner and the registrar for certificate-based authorization provides authorization of the specific device being deployed. Previously, introducer-to-petitioner device communication was secured only using physical security between the introducer and the petitioner device. SDP certificate-based authorization gives the registrar an opportunity to validate the current device identity before accepting the introduction.

**Authentication and Authorization Lists for SDP**

When you are configuring your SDP registrar, if you specify an authentication list and an authorization list, the registrar uses the specified lists for all introducer requests. The authentication list is used when authenticating the introducer (the AAA server checks for a valid account by looking at the username and password). The authorization list is used to receive the appropriate authorized fields for the certificate subject name and a list of template variables to be expanded into the Cisco IOS command-line interface (CLI) snippet that is sent
back to the petitioner. The authentication and authorization lists are usually point to the same AAA server list, but it is possible to use a different database for authentication and authorization. (Storing files on different databases is not recommended.)

When a petitioner makes an introduction request, multiple queries are sent to the AAA list database on the RADIUS or TACACS+ server. The queries search for entries of the following form:

```
user Password <userpassword>
cisco-avpair="tti:subjectname=<DN subjectname>>"
cisco-avpair="tti:iosconfig#<<value>>"
cisco-avpair="tti:iosconfig#<<value>>"
cisco-avpair="tti:iosconfig#=<<value>>"
```

The existence of a valid AAA username record is enough to pass the authentication check. The "cisco-avpair=tti" information is necessary only for the authorization check.

If a subject name was received in the authorization response, the SDP registrar stores it in the enrollment database, and that "subjectname" overrides the subject name that is supplied in the subsequent certificate request (PKCS10) from the petitioner device.

The numbered “tti:iosconfig” values are expanded into the SDP Cisco IOS snippet that is sent to the petitioner. The configurations replace any numbered ($1 through $9) template variable. Because the default Cisco IOS snippet template does not include the variables $1 through $9, these variables are ignored unless you configure an external Cisco IOS snippet template. To specify an external configuration, use the `template config` command. For more information on external configurations, see the section “Authentication and Authorization Lists for SDP, on page 154.”

The template configuration location may include a variable "$n," which is expanded to the name with which the user is logged in.

**Authentication and Authorization Lists for an Administrative Introducer**

The SDP mechanisms assume a permanent relationship between the introducer and the device. As a result, the introducer username is used to define the device name.

In some SDP deployment scenarios, the introducer is an administrator doing the introduction for many devices. However, using the introducer (the administrator) name to define the device name results in multiple devices being incorrectly deployed with the same device name. Instead, an administrative introducer allows the administrator to specify the correct device name during the introduction.

More generally stated, the introducer username is used as the database record locator to determine all other information about the device including the Cisco IOS configuration template, various template variables (pulled from an AAA database and expanded into the template), and the appropriate subject name for PKI certificates issued to the device. For simplicity, this database record locator is called the user/device name.

The administrative introducer provides a device name. In that way, an administrator can provide the appropriate record locator when doing an introduction. For example, if an administrator is trying to introduce a device for username “user1,” the administrator introduces the device into the PKI network and provides user1 as the record locator after logging into the registrar using the administrator’s own credentials. The record locator, user1, becomes the device name. All other template and PKI certificate subject name information specific to the introduction is then provided by the user1 username records instead of by the administrator’s record.
The registrar device uses the supplied username information with a user introducer name. The username allows the existing mechanisms for determining a user’s authorization, template, and PKI certificate information to be supported without modification.

### How Custom Templates Work with SDP

You may use custom templates to streamline the SDP process.

- Custom templates allow you to complete the web pages with the required start information, so the introducer is no longer required to contact the registrar and can immediately begin the SDP transaction.
- Custom templates allow customized deployment information to be displayed on the web pages, thereby tailoring the user experience.

An easy way to define a custom template is to modify the default template. Without custom templates, the introducer must contact the registrar for information to begin the SDP transaction. For a list of the default templates, see the section “Default Templates for SDP Transaction Web Pages, on page 160.”

#### Note

It is recommended that only advanced SDP users configure custom templates because problems can result from modifying templates incorrectly before the templates are displayed in the introducer’s browser.

### Custom Template Variable Expansion

There are expansion variables in the templates that are replaced by the Cisco IOS SDP registrar or petitioner. These variables are expanded as follows:

- $S$--“$”
- $a$--attribute-value (AV) pairs
- $c$--Trusted certificate
- $d$--Dump AV pairs in browser
- $h$--Hostname
- $k$--Keylabel or “tti”
- $l$--Trustpoint label = “tti”
- $n$--HTTP client’s username
- $s$--Default TTI key size
- $t$--Trustpoint configuration
- $u$--Completion URL
- $1$ to $9$--Variables retrieved from AAA server during user authentication
Custom Template Variable Expansion Rules

Configuration and templates are used during an SDP exchange. Prior to use and after distribution, these templates are expanded using the following rules based in the SDP communication stage.

Custom HTML Template Expansion Rules

HTML templates are expanded immediately before being served to the HTTP client. The HTTP templates are expanded as follows:

- $u--Completion url, which is be populated with the SDP completion URL (for example: http://10.10.10.1/ezsdd/completion ). This variable is used internally by SDP as the internal "wizard" state. It is expected that the SDP introduction page include something similar to the following text: "<FORM action="$u" method="post">" for normal wizard processing.
- $n--introducer name or the device name entered by the administrative introducer.
- $$--$
- $h--Hostname
- $a--All AV pairs with or without a specified template character are written in the following HTML form format. (Because these AV pairs are not "INPUT type=hidden," they are directly displayed on the web page for debugging templates or the SDP process.)

```html
<INPUT type=hidden NAME="attribute string here" value="variable string here"><BR>
all HTML templates should have this!
```

```
$\text{d = dump all av pairs in: attribute = value}<BR>
```

URL Template Expansion Rules

There are URLs for the configuration template source, the file template source, and the file destination. These variables are expanded when the registrar prepares the URL, just before retrieving the configuration or file. For the file destination, these variables are expanded just before the petitioner copies the file to the file destination.

- $$--$
- $h--Hostname

URL Template Expansion Rules for iPhone Deployment

The following template expansion variables are introduced for iPhone deployment:

- $o-challenge password. This template character is expanded by the SDP registrar after it obtains the challenge password from the Simple Certificate Enrollment Protocol (SCEP) server, before the configuration profile is sent to the iPhone in the START phase.
- $i-uniquedevice identifier (UDID) of the iPhone. This template character is expanded by the SDP registrar into the CN field of the Subject Name, before the configuration profile is sent to the iPhone in the INTRODUCTION phase.
• $p$ - subject name differentiator. This template character is expanded by the SDP registrar using the value configured through the CLI. See the Configuring the SDP Registrar to Deploy Apple iPhones, on page 177 for more information. This value can be used to differentiate the two certificates issued by the SCEP server to the iPhone, one in the COMPLETION phase and one in the VPN establishment phase. You determine part and field of the Subject Name into which this value goes.

See the How SDP Deploys Apple iPhones in a PKI, on page 163 for more information.

Custom Configuration and File Template Variable Expansion Rules

Custom configuration and file template variables are expanded both when the registrar prepares the configuration or file template and when the petitioner receives the configuration or file template.

Custom Configuration and File Template Variable Expansion Rules at the Registrar

When the registrar expands the configuration or file template, the following variables are used by the Cisco IOS CA. These variables are expanded before being sent through the SDP wizard.

• $$--$$
• $h$--Hostname
• $t$--A simple default trustpoint configuration that includes $l$, $k$, and $s$ to be expanded at the client
• $1$ to $9$--Variables retrieved from AAA server during user authentication (not applicable to the file template)

Custom Configuration and File Template Variable Expansion Rules at the Petitioner

When the petitioner expands the configuration or file template, the following variables are expanded:

• $$--$$
• $h$--Hostname
• $k$--Keylabel
• $l$--Trustpoint label
• $s$--Key size
• $c$--Expanded to certificate chain
• $n$--Expanded to username (not applicable to the file template)

Custom Configuration HTTP Template Variable Expansion Rules

Custom configuration HTTP templates provide flexibility for backend Common Gateway Interface (CGI) scripts and integration with external management systems. Template URLs run through the HTTP template expansions before registrar retrieves the bootstrap configuration from the external management system. The device name ($n$) is expanded into the URL and passed to the external management system so that a specific bootstrap configuration file can be located based on the device information.
You should only modify the HTML text that is displayed. The existing expansion variables, Javascript, and forms in the default templates should not be removed when customizing the templates. They are required for SDP to function properly.

The HTTP template expansion and `template config` command allow you to specify either of the following file types to obtain a customized bootstrap configuration file:

- A configuration file based on the device name (for example, `template config http://myserver/$(n-config-file.conf)
- A CGI script based on the device name (for example, `template config http://myserver/cgi-bin/mysdpcgi post`)

As of Cisco IOS Release 12.4(6)T, the CGI support has been expanded so that the bootstrap configuration can be identified by not only the device name, but also the type, current Cisco IOS version information, and current configuration. This functionality expands the `template config` command with the `post` keyword, which tells the registrar to send this additional device information to the external management system through a CGI script with the HTTP or HTTPS protocol only.

The registrar passes the device information through AV pairs ($a) to the external management system. Using the AV pair information, the management system identifies the appropriate bootstrap configuration file and sends it back to the registrar. The additional AV pairs that are sent with the expanded CGI support for identification of the customized bootstrap configuration file are shown in the table below.

### Table 5: AV Pairs Sent During HTTP Post to External Management System

<table>
<thead>
<tr>
<th>AV Pair</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTIFixSubjectName</td>
<td>AAA_AT_TTI_SUBJECTNAME (sent only if the realm authentication user is not the root user on the registrar)</td>
</tr>
<tr>
<td>TTIIosRunningConfig</td>
<td>Output of <code>show running-config brief</code></td>
</tr>
<tr>
<td>TTIKeyHash</td>
<td>Digest calculated over the device public key</td>
</tr>
<tr>
<td>TTIPrivilege</td>
<td>AAA_AT_TTI_PRIVILEGE--&quot;admin&quot; is sent if the user is an administrator, &quot;user&quot; is sent if the user is not an administrator (sent only if the realm authentication user is an administrator and the information is available from the AAA server)</td>
</tr>
<tr>
<td>TTISignature</td>
<td>Digest calculated over all AV pairs except UserDeviceName and TTISignCert</td>
</tr>
<tr>
<td>TTISignCert</td>
<td>Device current certificate (sent only if the device currently has a certificate)</td>
</tr>
<tr>
<td>TTTTemplateVar</td>
<td>AAA_AT_TTI_IOSCONFIG(1-9) (sent only if the realm authentication user is not the root user on the registrar)</td>
</tr>
</tbody>
</table>
### AV Pair

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTIUserName</td>
</tr>
<tr>
<td>Device name</td>
</tr>
<tr>
<td>TTIVersion</td>
</tr>
<tr>
<td>TTI version of the registrar</td>
</tr>
<tr>
<td>UserDeviceName</td>
</tr>
<tr>
<td>Device name as entered by the administrative introducer (sent only if the realm authentication user is an administrator)</td>
</tr>
</tbody>
</table>

**Note**

The registrar must be running Cisco IOS Release 12.4(6)T, the `template config` command must be issued with the `post` keyword, and the `url` argument must include either HTTP or HTTPS. No other protocol is supported for the expanded CGI template functionality (for example, FTP).

### Default Templates for SDP Transaction Web Pages

The following default templates exist for each SDP transaction web page:

- Default Prep-Connect Template, on page 160
- Default Start Page Template, on page 162
- Default Welcome Page Template, on page 162
- Default Introduction Page Template, on page 162
- Default Admin-Introduction Page Template, on page 162
- Default Completion Page Template, on page 163

### Default Prep-Connect Template

The prep-connect template may be modified by the administrator to contain values that are appropriate for their environment. The format of the prep-connect page may also be modified by the settings contained in the template.

Except for the registrar IP address, which the administrator must customize, the prep-connect template may be used as shown below.

```html
<html>
<head>
<title>SDP: Test Internet Connection</title></head>
<noscript>
If you see this message, your browser is not running JavaScript, which is required by Cisco Secure Device Provisioning.<br>
If you cannot enable JavaScript, please contact your system administrator.
</noscript>
<body style="background-color: rgb(204, 255, 255);">
<form action="http://10.10.10.1/ezsdd/connect" method="post">
<input type="submit" value="Log onto Cisco Device"/>
</form>
</body>
</html>
```

Default username/password is cisco/cisco.
Hidden HTML Form Fields

The hidden HTML form fields communicate initial configuration information to the browser as set by the administrator and are not signed.

The term "hidden" refers to the fact that these HTML form fields are not displayed on the prep-connect page to reduce potential confusion to the introducer.

The administrator can set hidden HTML form fields in the prep-connect template as shown in the table below.

### Table 6: Administrator Defined AV Pairs Sent During Prep-Connect Phase

<table>
<thead>
<tr>
<th>AV Pair</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTIAfterConnectURL</td>
<td>The administrator may set the TTIAfterConnectURL field to either the welcome page URL or the start page URL. The welcome page URL is specified with the factory default petitioner IP address. The connect after URL may be any valid URL if SDP is not going to be used after establishing Internet connectivity.</td>
</tr>
<tr>
<td>TTIConnectTestURL</td>
<td>The administrator may set the TTIConnectTestURL field to a valid URL that should be accessible when Internet connectivity is established. The default prep-connect template value is <a href="http://www.cisco.com">www.cisco.com</a> (198.133.219.25).</td>
</tr>
<tr>
<td>TTIInsideAddr</td>
<td>The administrator may set the TTIInsideAddr field to the factory default IP address of the petitioner. For the Cisco 871 ISR, the IP address is 10.10.10.1.</td>
</tr>
<tr>
<td>TTIlanportx</td>
<td>The administrator may set the TTIlanportx field to the LAN interface name of the petitioner platform. This field is used to apply the Cisco IOS connect configuration. For the Cisco 871, the field value is “Vlan1.”</td>
</tr>
<tr>
<td>TTIwanport</td>
<td>The administrator may set the TTIwanport field to the WAN interface name of the petitioner. This field is used to apply the Cisco IOS connect configuration. For the Cisco 871, the field value is “FastEthernet4.”</td>
</tr>
</tbody>
</table>
The connect template cannot be customized.
How SDP Deploys Apple iPhones in a PKI

With the introduction of the Cisco IOS 15.1(2)T and Apple iPhone OS 3.0 releases, Apple iPhones are supported on Cisco IOS network devices. Cisco IOS routers use the SDP registrar to deploy iPhones so that network applications can be accessed securely through an IPSec VPN, SCEP server, and PKI certificate deployment technologies.

The Apple iPhone combines the distribution of its XML-based “Configuration Profiles” with the initial deployment of certificates. SDP uses these initial certificates to authenticate access to enterprise applications.
and encrypt subsequent profile distribution. SDP uses this enrollment solution for distributing digital certificates to the iPhone.

**Figure 17: SDP Registrar Deployment of the iPhone in a PKI**

The following sections describe each phase of the SDP registrar deployment of the iPhone in a PKI:

**SDP Registrar Deployment Phases of the Apple iPhone in a PKI**

The following sections describe each phase of the SDP registrar deployment of the iPhone in a PKI:

**Start SDP Deployment Phase**

The following steps describe the Start SDP deployment phase:

---

**Note**


---

**SUMMARY STEPS**

1. The iPhone user opens the Safari browser and types the start page HTTPS URL. For example, this HTTPS URL may be an internal corporate network address. The SDP registrar HTTPS page initiates the process.
2. The user starts authentication with the Cisco router, which acts as the SDP registrar by providing a username and password.
3. The SDP registrar contacts the SCEP server to obtain a challenge password.
4. The SDP registrar constructs a configuration profile in XML format that consists of the challenge password, SCEP server URL, and a request for iPhone attributes. The SCEP server URL is used to send the enrollment request and the iPhone device attributes are used by the iPhone to generate the RSA keys.
5. The iPhone user installs the configuration profile on the iPhone to complete the Start SDP phase.
DETAILED STEPS

Step 1  The iPhone user opens the Safari browser and types the start page HTTPS URL. For example, this HTTPS URL may be an internal corporate network address. The SDP registrar HTTPS page initiates the process.

Step 2  The user starts authentication with the Cisco router, which acts as the SDP registrar by providing a username and password.

Step 3  The SDP registrar contacts the SCEP server to obtain a challenge password.

Step 4  The SDP registrar constructs a configuration profile in XML format that consists of the challenge password, SCEP server URL, and a request for iPhone attributes. The SCEP server URL is used to send the enrollment request and the iPhone device attributes are used by the iPhone to generate the RSA keys. The following example shows a configuration profile sent by the SDP registrar to the iPhone in the Start SDP deployment phase:

Example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Inc//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>PayloadContent</key>
<dict>
<key>URL</key>
<string>https://profileserver.example.com/iphone</string>
<key>DeviceAttributes</key>
<array>
<string>UDID</string>
<string>IMEI</string>
<string>ICCID</string>
<string>VERSION</string>
<string>PRODUCT</string>
</array>
<key>Challenge</key>
<string>optional challenge</string>
</dict>
</dict>
</plist>
```

Step 5  The iPhone user installs the configuration profile on the iPhone to complete the Start SDP phase.

Welcome SDP Deployment Phase

The Welcome SDP deployment phase is not applicable for the iPhone because the Introducer (for example, Safari web browser) is run on the SDP petitioner (iPhone).

Introduction SDP Deployment Phase

The following steps describe the Introduction SDP deployment phase:

### SUMMARY STEPS

1. The iPhone triggers an HTTPS post containing the requested device attribute information and the challenge password as a configuration profile. The HTTPS post is directed to the HTTPS URL specified in the configuration profile obtained in the Start SDP deployment phase, which must be the Introduction SDP deployment phase URL. The post data is signed by the iPhone using an Apple-issued certificate (built-in identity) and this signature may be verified, the identity confirmed, and the device attributes checked.

2. The UDID sent by the iPhone is captured by the SDP registrar and included in the Subject Name. Going forward, the device attributes obtained by the SDP registrar are used to determine if this was exactly the type of device that would be accepted. For example, the network administrator would only let 3GS iPhones onto the network because they have hardware encrypted storage. The device attributes obtained would enable the SDP registrar to distinguish 3GS iPhones from 3G iPhones.

3. The SDP registrar responds by building a configuration profile that consists of the following: HTTP URL of the SCEP server, Subject Name (contains the UDID) that is sent in the enrollment request, key size, key type, key usage, and challenge password. If the START phase had been skipped, the SDP registrar would contact the SCEP server to obtain a challenge password. See the URL Template Expansion Rules for iPhone Deployment, on page 157 for more information about how the SDP registrar obtains the Subject Name and the challenge password.

### DETAILED STEPS

#### Step 1

The iPhone triggers an HTTPS post containing the requested device attribute information and the challenge password as a configuration profile. The HTTPS post is directed to the HTTPS URL specified in the configuration profile obtained in the Start SDP deployment phase, which must be the Introduction SDP deployment phase URL. The post data is signed by the iPhone using an Apple-issued certificate (built-in identity) and this signature may be verified, the identity confirmed, and the device attributes checked.

#### Step 2

The UDID sent by the iPhone is captured by the SDP registrar and included in the Subject Name. Going forward, the device attributes obtained by the SDP registrar are used to determine if this was exactly the type of device that would be accepted. For example, the network administrator would only let 3GS iPhones onto the network because they have hardware encrypted storage. The device attributes obtained would enable the SDP registrar to distinguish 3GS iPhones from 3G iPhones.

The following example shows a configuration profile sent by the iPhone in the Introduction SDP deployment phase:

**Example:**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>UDID</key>
    <string></string>
    <key>VERSION</key>
    <string>7A182</string>
    <key>MAC_ADDRESS_EN0</key>
    <string>00:00:00:00:00:00</string>
    <key>CHALLENGE</key>
    <string>either:
      <string>String</string>
    or:
      <data>base64 encoded data</data>
  </dict>
</plist>
```
Step 3  
The SDP registrar responds by building a configuration profile that consists of the following: HTTP URL of the SCEP server, Subject Name (contains the UDID) that is sent in the enrollment request, key size, key type, key usage, and challenge password. If the START phase had been skipped, the SDP registrar would contact the SCEP server to obtain a challenge password. See the URL Template Expansion Rules for iPhone Deployment, on page 157 for more information about how the SDP registrar obtains the Subject Name and the challenge password.

**Note**  
The SDP registrar supports the RSA key type only.

The following example shows a configuration profile sent by the SDP registrar in the Introduction SDP deployment phase:

**Example:**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Inc//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>PayloadContent</key>
    <dict>
      <key>URL</key>
      <string>https://iphone.vpn.apple.com/pkifoobar.exe</string>
      <key>Name</key>
      <string>instance_for_getcacert_call</string>
      <key>Subject</key>
      <array>
        <array>
          <array>
            <string>O</string>
            <string>Apple Inc.</string>
          </array>
        </array>
        <array>
          <array>
            <string>CN</string>
            <string>Foo</string>
          </array>
        </array>
      </array>
      <key>Challenge</key>
      <string>CHALLENGE</string>
      <key>Keysize</key>
      <integer>1024</integer>
      <key>KeyType</key>
      <string>RSA</string>
      <key>KeyUsage</key>
      <integer>5</integer>
    </dict>
    <key>PayloadDescription</key>
    <string>Provides device encryption identity</string>
    <key>PayloadUUID</key>
    <string>fd8a6b9e-0fed-406f-9571-8ec98722b713</string>
    <key>PayloadType</key>
    <string>com.apple.security.scep</string>
    <key>PayloadDisplayName</key>
    <string>Encryption Identity</string>
    <key>PayloadVersion</key>
    <integer>1</integer>
    <key>PayloadOrganization</key>
    <string>Apple Inc.</string>
    <key>PayloadIdentifier</key>
    <string>com.apple.encrypted-profile-service</string>
  </dict>
</plist>
```
Post-Introduction SDP Deployment Phase

The following steps describe the Post-introduction SDP deployment phase.

**Note**

**SUMMARY STEPS**

1. The iPhone installs the configuration profile specification containing SCEP information obtained from the SDP registrar in the Introduction SDP deployment phase.
2. The iPhone generates the keys with the instructions in the profile specification and sends the enrollment request to the SCEP server whose HTTP URL is specified in the profile, along with the challenge password.
3. The SCEP server verifies the challenge password and issues the digital certificate to the iPhone.
4. The user can install this certificate on the iPhone and use the Cisco IPsec VPN to connect to the corporate network.

**DETAILED STEPS**

**Step 1**
The iPhone installs the configuration profile specification containing SCEP information obtained from the SDP registrar in the Introduction SDP deployment phase.

**Step 2**
The iPhone generates the keys with the instructions in the profile specification and sends the enrollment request to the SCEP server whose HTTP URL is specified in the profile, along with the challenge password.

**Step 3**
The SCEP server verifies the challenge password and issues the digital certificate to the iPhone.

**Step 4**
The user can install this certificate on the iPhone and use the Cisco IPsec VPN to connect to the corporate network.

**Note**
This certificate can also be used to download other enterprise settings, such as VPN settings, and Wi-Fi settings.

Second-Introduction SDP Deployment Phase

The following steps describe the Second-introduction SDP deployment phase:

**Note**
SUMMARY STEPS

1. The iPhone repeats the Introduction SDP deployment phase with the following exceptions:
2. The SDP registrar responds with a configuration profile that includes the general enterprise settings such as VPN settings, Wi-Fi settings, and email settings, and in addition includes SCEP settings for a second certificate to be used for establishing a VPN.

DETAILED STEPS

Step 1
The iPhone repeats the Introduction SDP deployment phase with the following exceptions:

• The iPhone does not include the challenge password as part of the post data.
• The iPhone signs the post data using the certificate obtained from the SCEP server in the Post-introduction SDP deployment phase.

Step 2
The SDP registrar responds with a configuration profile that includes the general enterprise settings such as VPN settings, Wi-Fi settings, and email settings, and in addition includes SCEP settings for a second certificate to be used for establishing a VPN.

Second Post-Introduction SDP Deployment Phase
The Second Post-introduction SDP phase is identical to the Post-introduction SDP deployment phase. The iPhone generates a certificate request based on the SCEP settings provided by the SDP registrar in the Second-introduction SDP deployment phase and enrolls with the SCEP server.

Completion SDP Deployment Phase
The Completion SDP deployment phase is not applicable for the iPhone because the Introducer (for example, the Safari web browser) is run on the SDP petitioner (iPhone).

How to Set Up Secure Device Provisioning (SDP) for Enrollment in a PKI

This section contains the following procedures that should be followed when setting up SDP for your PKI. You can configure the registrar according to only one of the registrar configuration tasks.

Enabling the SDP Petitioner
Perform this task to enable or disable the petitioner and associate a trustpoint with the SDP exchange. You can also use this task to configure the petitioner to use a certificate and the RSA keys associated with a specific trustpoint.
Enabling the SDP Petitioner

The petitioner is enabled by default on a Cisco device that contains a crypto image; thus, you have only to issue the `crypto provisioning petitioner` command if you have previously disabled the petitioner or if you want to use an existing trustpoint instead of the automatically generated trustpoint.

Note

By default, the SDP petitioner device uses an existing certificate. If multiple certificates and one specific certificate exist, use this task to make a choice. However, this task is not necessary to enable the default behavior.

Before You Begin

- The HTTP server must be enabled through the `ip http server` command. (The HTTP server is typically enabled by default in many default Cisco IOS configurations.)
- If you are configuring the petitioner to use a certificate and RSA keys, your SDP petitioner device must have an existing manufacturer's certificate or a third-party certificate.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `crypto provisioning petitioner`
4. Do one of the following:
   - `trustpoint trustpoint-label`
5. `end`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router&gt; enable</code></td>
<td></td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
</tr>
</tbody>
</table>
| 3    | crypto provisioning petitioner | Allows SDP petitioner device behavior to be modified and enters tti-petitioner configuration mode.  
*Note* Effective with Cisco IOS Release 12.3(14)T, the `crypto provisioning petitioner` command replaced the `crypto wui tti petitioner` command. |
| 4    | Do one of the following:  
• `trustpoint trustpoint-label` | (Optional) Specifies the trustpoint that is to be associated with the SDP exchange between the petitioner and the registrar.  
*Note* If this command is not issued, the `trustpoint-label` argument is automatically labeled "tti." (Optional) Specifies the trustpoint and associated certificate that are used when signing all introduction data during the SDP exchange. |
| 5    | `end` | (Optional) Exits tti-petitioner configuration mode. |

**Troubleshooting Tips**

After the SDP exchange is complete, a new trustpoint-label named "tti" exists. The trustpoint is automatically enrolled with the certificate server (the registrar). To verify that the trustpoint is really there, use the `show running-config` command.
What to Do Next

If you set up the petitioner to use a certificate and the RSA keys associated with the specified trustpoint, you should configure the registrar as shown in the task "Enabling the SDP Registrar for Certificate-Based Authorization."

Enabling the SDP Registrar and Adding AAA Lists to the Server

Perform this task to enable the registrar and associate a certificate server with the SDP exchange.

You can also use this task if you want to add an authentication list and an authorization list to the RADIUS or TACACS+ server.

Prerequisites

Before configuring a registrar, perform the following tasks:

• Enable the HTTP server or the HTTPS server.

Note

Before you enable an HTTPS server, you must disable the standard HTTP server if it is configured. Use the the no ip http server command to disable an HTTP server. To enable an HTTPS server, you should issue the ip http secure-server command followed by the ip http secure-trustpoint command. The specified trustpoint is a registrar local trustpoint appropriate for HTTPS communication between the registrar and the user’s browser.

• Configure the Cisco IOS certificate server through the crypto pki server command.

If you are configuring AAA lists, you should complete the prerequisites required for the registrar in addition to completing the following tasks:

• Add user information to the AAA server database. To configure a RADIUS or TACACS+ AAA server, see the “Configuring RADIUS” and “Configuring TACACS+” chapters of the Cisco IOS Security Configuration Guide.


Restrictions

Cisco IOS CA Device Requirement

During the SDP process, a Cisco IOS CA certificate is automatically issued to the peer device. If an SDP registrar is configured on a third-party vendor’s CA device, the SDP process does not work.
The template config Command

There are nine Cisco IOS configuration variables. If you require more configuration flexibility, the template config command can be used to reference a configuration template that is specific to the introducer. For more information on configuration flexibility, see the "Custom Configuration and File Template Variable Expansion Rules, on page 158" section.

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto provisioning registrar
4. pki-server label
5. authentication list list-name
6. authorization list list-name
7. template username name password password
8. template config url [post]
9. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Step 3 crypto provisioning registrar</td>
<td>Configures a device to become a registrar for the SDP exchange and enters tti-registrar configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# crypto provisioning registrar</td>
</tr>
<tr>
<td>Note</td>
<td>Effective with Cisco IOS Release 12.3(14)T, the crypto provisioning registrar command replaced the crypto wui tti registrar command.</td>
</tr>
<tr>
<td>Step 4 pki-server label</td>
<td>Specifies the certificate server that is to be associated with the SDP exchange between the petitioner and the registrar.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(tti-registrar)# pki-server mycs</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
</tbody>
</table>
| 5    | **authentication list** `list-name` | (Optional) Authenticates the introducer in an SDP exchange.  
**Example:**  
Router (tti-registrar)# authentication list authen-tac |
| 6    | **authorization list** `list-name` | (Optional) Receives the appropriate authorized fields for the certificate subject name and list of template variables to be expanded into the Cisco IOS CLI snippet that is sent back to the petitioner.  
**Example:**  
Router (tti-registrar)# authorization list author-rad |
| 7    | **template username** `name` **password** `password` | (Optional) Establishes a username and password in which to access the configuration template on the file system.  
**Example:**  
Router (tti-registrar)# template username ftpuser password ftppwd |
| 8    | **template config** `url [post]` | (Optional) Specifies a remote URL for the Cisco IOS CLI configuration template.  
**Example:**  
Router (tti-registrar)# template config http://myserver/cgi-bin/mycgi post |
| 9    | **end** | (Optional) Exits tti-registrar configuration mode.  
**Example:**  
Router (tti-registrar)# end |

**Examples**

To help troubleshoot the SDP transaction, you can issue the `debug crypto provisioning` command, which displays output from the petitioner and registrar devices.
The following is output for the `debug crypto provisioning` command. The output from the petitioner and registrar devices are shown below.

Petitioner device
! The user starts the Welcome phase.
Nov 7 03:15:48.171: CRYPTO_PROVISIONING: received welcome get request.
! The router generates an Rivest, Shamir, and Adelman (RSA) keypair for future enrollment.
Nov 7 03:15:48.279: CRYPTO_PROVISIONING: keyhash 'A506BE3B83C6F4B4A6EFCEB3D584AA4A'
! The TTI transaction is completed.
Nov 7 03:16:10.607: CRYPTO_PROVISIONING: received completion post request.

Registrar device
! During the introduction phase, the browser prompts for login information.
06:39:18: CRYPTO_PROVISIONING: received introduction post request.
06:39:18: CRYPTO_PROVISIONING: checking AAA authentication (ipsecca_script_aalists, ttiuser)
! This happens if the user types in the wrong username or password.
06:39:19: CRYPTO_PROVISIONING: authentication declined by AAA, or AAA server not found - 0x3
06:39:19: CRYPTO_PROVISIONING: aaa query fails!
! The user re-enters login information.
06:39:19: CRYPTO_PROVISIONING: received introduction post request.
06:39:19: CRYPTO_PROVISIONING: checking AAA authentication (ipsecca_script_aalists, ttiuser)
06:39:20: CRYPTO_PROVISIONING: checking AAA authorization (ipsecca_script_aalists, ttiuser)
! The login attempt succeeds and authorization information is retrieved from the AAA database.
06:39:21: CRYPTO_PROVISIONING: aaa query ok!
! These attributes are inserted into the configuration template.
06:39:21: CRYPTO_PROVISIONING: building TTI av pairs from AAA attributes
06:39:21: CRYPTO_PROVISIONING: "subjectname" = "CN=user1, O=company, C=US"
06:39:21: CRYPTO_PROVISIONING: "$1" = "ntp server 10.3.0.1"
06:39:21: CRYPTO_PROVISIONING: "$2" = "hostname user1-vpn"
! The registrar stores this subject name and overrides the subject name in the subsequent enrollment request.
06:39:21: CRYPTO_PROVISIONING: subjectname=CN=user1, O=company, C=US
! The registrar stores this key information so that it may be used to automatically grant the subsequent enrollment request.
06:39:21: CRYPTO_PROVISIONING: key_hash=A506BE3B83C6F4B4A6EFCEB3D584AA4A

Enabling the SDP Registrar for Certificate-Based Authorization

Perform this task to enable the SDP registrar to verify the petitioner-signing certificate using either a specified trustpoint or any configured trustpoint and initiate authorization lookups using the introducer username and the certificate name field.

Before You Begin
You must also configure the SDP petitioner to use a certificate and RSA keys associated with a specific trustpoint. To complete this task, use the trustpoint signing command as shown in the task “Enabling the SDP Petitioner, on page 169.”

> Note

Because RADIUS does not differentiate between authentication and authorization, you need to use the default password, cisco, for certificate authorization.
### SUMMARY STEPS

1. enable
2. configure terminal
3. crypto provisioning registrar
4. template file `sourceURL destinationURL`
5. binary file `sourceURL destinationURL`
6. authentication trustpoint `{trustpoint-label| use-any }
7. authorization `{login | certificate | login certificate`
8. authorization username subjectname `subjectname`
9. end

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
• Enter your password if prompted.  
  **Example:**  
  `Router> enable`
| **Step 2** configure terminal | Enters global configuration mode.  
  **Example:**  
  `Router# configure terminal`
| **Step 3** crypto provisioning registrar | Configures a device to become an SDP registrar and enters tti-registrar configuration mode.  
  **Example:**  
  `Router(config)# crypto provisioning registrar`
| **Step 4** template file `sourceURL destinationURL` | (Optional) Specifies the source template file location on the registrar and the destination template file location on the petitioner.  
  **Note** This command is useful when using a USB token to provision a device.  
  The template expansion occurs on the registrar for both the source URL and file content. The destination URL is expanded on the petitioner.  
  **Example:**  
  `Router(tti-registrar)# template file http://myserver/registrar_file_r1 http://myserver/petitioner_file_p1`
| **Step 5** binary file `sourceURL destinationURL` | (Optional) Specifies the binary file location on the registrar and the destination binary file location on the petitioner.  
  **Note** This command is useful when using a USB token to provision a device.  
  Both the source and destination URL are expanded on the registrar.  
  Also, the destination URL and file content are expanded on the petitioner.  
  Binary files are not processed through the template expansion functions.  
  **Example:**  
  `Router(tti-registrar)# binary file`
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://myserver/registrar_file_a1">http://myserver/registrar_file_a1</a></td>
<td>(Optional) Specifies the trustpoint used to authenticate the SDP petitioner device’s existing certificate.</td>
</tr>
<tr>
<td><a href="http://myserver/petitioner_file_b1">http://myserver/petitioner_file_b1</a></td>
<td>• trustpoint-label --Specifies a specific trustpoint.</td>
</tr>
<tr>
<td></td>
<td>• use-any --Specifies any configured trustpoint.</td>
</tr>
<tr>
<td>Step 6 authentication trustpoint {trustpoint-label</td>
<td>use-any }</td>
</tr>
<tr>
<td>Example: Router(tti-registrar)# authentication trustpoint mytrust</td>
<td></td>
</tr>
<tr>
<td>Step 7 authorization {login</td>
<td>certificate</td>
</tr>
<tr>
<td>Example: Router(tti-registrar)# authorization login certificate</td>
<td>• Use the login keyword for authorization based on the introducer’s username.</td>
</tr>
<tr>
<td>Step 8 authorization username subjectname subjectname</td>
<td>Sets parameters for the different certificate fields that are used to build the AAA username.</td>
</tr>
<tr>
<td>Example: Router(tti-registrar)# authorization username subjectname all</td>
<td>• The all keyword specifies that the entire subject name if the certificate is used as the authorization username.</td>
</tr>
<tr>
<td>Step 9 end</td>
<td>(Optional) Exits tti-registrar configuration mode.</td>
</tr>
<tr>
<td>Example: Router(tti-registrar)# end</td>
<td></td>
</tr>
</tbody>
</table>

**Configuring the SDP Registrar to Deploy Apple iPhones**

Perform this task to configure the SDP registrar to run HTTPS in order to deploy Apple iPhones on a corporate network.

**Before You Begin**

Ensure that the SDP Registrar is enabled to run HTTPS. See the Enabling the SDP Registrar and Adding AAA Lists to the Server section for more information.
SUMMARY STEPS

1. enable
2. configure terminal
3. ip http secure-server
4. crypto provisioning registrar
5. url-profile start profile-name
6. url-profile intro profile-name
7. match url url
8. match authentication trustpoint trustpoint-name
9. match certificate certificate-map
10. mime-type mime-type
11. template location location
12. template variable p value

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip http secure-server</td>
<td>Enables the HTTPS web server.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# ip http secure-server</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> crypto provisioning registrar</td>
<td>Configures a device to become a registrar for the SDP exchange and enters tti-registrar configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# crypto provisioning registrar</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> Effective with Cisco IOS Release 12.3(14)T, the crypto provisioning registrar command replaced the crypto wui tti registrar command.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> url-profile start profile-name</td>
<td>Specifies the start keyword to indicate that a URL profile is to be associated with the Start SDP deployment phase. The profile-name argument specifies the name of a unique URL profile.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(tti-registrar)# url-profile start START</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Note</strong> Both the Introduction SDP deployment phase and the Start SDP deployment phase can use different profiles or use the same URL profile.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> url-profile intro profile-name</td>
<td>Specifies the <code>intro</code> keyword to indicate that a URL profile is to be associated with the Introduction SDP deployment phase. The <code>profile-name</code> argument specifies the name of a unique URL profile. Note Both the Introduction SDP deployment phase and the Start SDP deployment phase can use different profiles or use the same URL profile.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(tti-registrar)# url-profile intro INTRO</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> match url url</td>
<td>Specifies the URL to be associated with the URL profile.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(tti-registrar)# match url /sdp/intro</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> match authentication trustpoint trustpoint-name</td>
<td>(Optional) Specifies the trustpoint name that should be used to authenticate the peer's certificate. If the trustpoint name is not specified, then the trustpoint configured using the <code>authentication trustpoint command</code> in tti-registrar configuration mode is used to authenticate the peer's certificate. See the Enabling the SDP Registrar for Certificate-Based Authorization section for more information.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(tti-registrar)# match authentication trustpoint apple-tp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> match certificate certificate-map</td>
<td>(Optional) Specifies the name of the certificate map used to authorize the peer's certificate.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(tti-registrar)# match certificate cat 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> mime-type mime-type</td>
<td>Specifies the Multipurpose Internet Mail Extensions (MIME) type that the SDP registrar should use to respond to a request received through this URL profile.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(tti-registrar)# mime-type application/x-apple-aspen-config</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> template location location</td>
<td>Specifies the location of the template that the SDP Registrar should use while responding to a request received through this URL profile.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(tti-registrar)# template location flash:intro.mobileconfig</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> template variable p value</td>
<td>(Optional) Specifies the value that goes into the Organizational Unit (OU) field of the subject name in the trustpoint certificate to be issued by the SDP Registrar. See this field in the certificate presented in the Apple CA Server Trustpoint Certificate Configuration Example section below.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(tti-registrar)# template variable p iphone-vpn</td>
<td></td>
</tr>
</tbody>
</table>
Apple CA Server Trustpoint Certificate Configuration

The SDP Registrar must verify the signature generated from the iPhone’s trustpoint certificate in order to trust the Apple CA server certificate. The iPhone signs its messages using the trustpoint certificate, which is issued by Apple’s CA server during the Introduction SDP deployment phase.

The following example shows how to configure certificate enrollment using the manual cut-and-paste enrollment method of the Apple CA certificate:

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>See also the “How to Configure Certificate Enrollment for a PKI” section in the Configuring Certificate Enrollment for a PKI feature module for more detailed information about configuring a trustpoint certificate.</td>
</tr>
</tbody>
</table>

SUMMARY STEPS

1. The `crypto pki trustpoint` command is entered in global configuration mode to declare the trustpoint and a given name and enters ca-trustpoint configuration mode:

   Example:
   ```
   Router(config)# crypto pki trustpoint apple-tp
   ```

2. The `enrollment terminal` command is entered to specify manual cut-and-paste certificate enrollment

3. The `crypto pki authenticate` command retrieves the CA certificate and authenticates it from the specified TFTP server.

4. Copy the following block of text containing the base 64 encoded Apple CA trust certificate and paste it at the prompt.

5. The `exit` command is used to exit ca-trustpoint configuration mode and enter global configuration mode.

6. The `crypto provisioning registrar` command is entered in global configuration mode to specify the router to become a registrar for the SDP exchange and enters tti-registrar configuration mode.

7. The `url-profile command with the intro` keyword is entered in tti-registrar configuration mode to specify the unique URL profile name that is associated with the Introduction SDP deployment phase.

8. The `match authentication trustpoint` command is entered in tti-registrar configuration mode to specify the trustpoint name that should be used to authenticate the peer’s certificate.

DETAILED STEPS

Step 1

The `crypto pki trustpoint` command is entered in global configuration mode to declare the trustpoint and a given name and enters ca-trustpoint configuration mode:

Example:

```
Router(config)# crypto pki trustpoint apple-tp
```
Example:

Router(ca-trustpoint)# enrollment terminal

**Step 3**

**The crypto pki authenticate** command retrieves the CA certificate and authenticates it from the specified TFTP server.

Example:

Router(ca-trustpoint)# crypto pki authenticate apple-tp

**Step 4**

Copy the following block of text containing the base 64 encoded Apple CA trust certificate and paste it at the prompt.

Example:

```
I Bag Attributes

localKeyID: 7C 29 15 15 12 C9 CF F6 15 2B 5B 25 70 3D A7 9A 98 14 36 06
subject=/C=US/O=Apple Inc./OU=Apple iPhone/CN=Apple iPhone Device CA
issuer=/C=US/O=Apple Inc./OU=Apple Certification Authority/CN=Apple iPhone Certification Authority
-----BEGIN CERTIFICATE-----
MIIDaTCCAlGgAwIBAgIBATANBgkqhkiG9w0BAQUFADB5MQswCQYDVQQGEwJVUzET
MBEGA1UEChMKQXBwbGUwdGhlMjAyMRIaGBQwDQYJKoZIhvcNAQc4DQwMCQwNzAv
BhC3QGJ2h0dHA6Ly93d3cuYXBwcGxlY2EvaXBob25lLmNybDANBgkqhkiG9w0BAQUF
-----END CERTIFICATE-----
```

**Step 5**

The **exit** command is used to exit ca-trustpoint configuration mode and enter global configuration mode.

Example:

Router(ca-trustpoint)# exit

**Step 6**

**The crypto provisioning registrar** command is entered in global configuration mode to specify the router to become a registrar for the SDP exchange and entersitti-registrar configuration mode.

Example:

Router(config)# crypto provisioning registrar

**Step 7**

**The url-profile command with the intro** keyword is entered initti-registrar configuration mode to specify the unique URL profile name that is associated with the Introduction SDP deployment phase.
Configuring an Administrative Introducer

Perform the following task to configure an administrative introducer using administrator authentication and authorization lists.

**Before You Begin**

The administrative introducer must have enable privileges on the client device and administrator privileges on the server.

**Note**

When using RADIUS, a user/device that needs to be introduced by the administrative introducer must always use cisco as its own password. TACACS+ does not have this limitation; a user/device can have any password and be introduced by the administrative introducer.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. crypto provisioning registrar
4. administrator authentication list *list-name*
5. administrator authorization list *list-name*
6. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;Router&gt; enable</td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong>&lt;br&gt;<strong>configure terminal</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong>&lt;br&gt;<strong>crypto provisioning registrar</strong></td>
<td>Configures a device to become an SDP registrar and enters tti-registrar configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;Router(config)# crypto provisioning registrar</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong>&lt;br&gt;<strong>administrator authentication list list-name</strong></td>
<td>Configures the AAA list used to authenticate an administrator during an introduction.</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;Router(tti-registrar)# administrator authentication list authen-tac</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong>&lt;br&gt;<strong>administrator authorization list list-name</strong></td>
<td>Configures the AAA list used to obtain authorization information for an administrator during an introduction. Information that can be obtained includes the certificate subject name and/or the list of template variables to be expanded into the Cisco IOS CLI snippet that is sent back to the petitioner.</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;Router(tti-registrar)# administrator authorization list author-tac</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong>&lt;br&gt;<strong>end</strong></td>
<td>(Optional) Exits tti-registrar configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong>&lt;br&gt;Router(tti-registrar)# end</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

The following example from the `show running-config` command allows you to verify that an administrative introducer using administrator authentication and authorization lists have been created:

```
Router# show running-config
Building configuration...
Current configuration : 2700 bytes
!
! Last configuration change at 01:22:26 GMT Fri Feb 4 2005
!
version 12.4
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
```
! hostname router
!
boot-start-marker
boot-end-marker

memory-size iomem 5
enable secret 5 $1$tpBS$PXnBDTIDXfX5pWa//1JX20
enable password lab
!
aaa new-model
!
!
aaa session-id common
!
resource manager
!
clock timezone GMT 0
ip subnet-zero
no ip routing
!
no ip dhcp use vrf connected
!
no ip cef
no ip domain lookup
ip domain name company.com
ip host router 10.3.0.6
ip host router.company.com 10.3.0.6
no ip ips deny-action ips-interface
!
no ftp-server write-enable
!
crypto pki server mycs

crypto pki trustpoint mycs
  revocation-check crl
  rsakeypair mycs
!
crypto pki trustpoint tti
  revocation-check crl
  rsakeypair tti
!
crypto pki trustpoint mic
  enrollment url http://router:80
  revocation-check crl
!
crypto pki trustpoint cat
  revocation-check crl
!
!
crypto pki certificate map cat 10
!
crypto pki certificate chain mycs
  certificate ca 01
crypto pki certificate chain tti
crypto pki certificate chain mic
  certificate 02
  certificate ca 01
crypto pki certificate chain cat

crypto provisioning registrar <-------- !SDP registrar device parameters!
  administrator authentication list authen-tac
  administrator authorization list author-tac
!
no crypto engine onboard 0
username qa privilege 15 password 0 lab
Configuring Custom Templates

Perform this task to create and configure custom templates.

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto provisioning registrar
4. template http start URL
5. template http welcome URL
6. template http introduction URL
7. template http admin-introduction URL
8. template http completion URL
9. template http error URL
10. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto provisioning registrar</td>
<td>Configures a device to become an SDP registrar and enters tti-registrar configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# crypto provisioning registrar</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> template http start URL</td>
<td>Directs the TTI registrar to use the custom start page template.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(tti-registrar)# template http start tftp://registrar.company .com/start.html</td>
<td>Note: This command is required to use the start page functionality. If this command is not issued, the welcome page is the initial communication between the introducer and the petitioner.</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>5</td>
<td>template http welcome $URL$</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(tti-registrar)# template http welcome tftp://registrar.company.com/welcome.html</td>
</tr>
<tr>
<td>6</td>
<td>template http introduction $URL$</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(tti-registrar)# template http introduction tftp://registrar.company.com/intro.html</td>
</tr>
<tr>
<td>7</td>
<td>template http admin-introduction $URL$</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(tti-registrar)# template http admin-introduction tftp://registrar.company.com/admin-intro.html</td>
</tr>
<tr>
<td>8</td>
<td>template http completion $URL$</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(tti-registrar)# template http completion tftp://registrar.company.com/completion.html</td>
</tr>
<tr>
<td>9</td>
<td>template http error $URL$</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(tti-registrar)# template http error tftp://registrar.company.com/error.html</td>
</tr>
<tr>
<td>10</td>
<td>end</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(tti-registrar)# end</td>
</tr>
</tbody>
</table>

**Example**

The following example shows the use of custom start, introduction, and completion templates:

```
template http start tftp://registrar.company.com/start.html
template http introduction tftp://registrar.company.com/intro.html```

---

**Setting Up Secure Device Provisioning for Enrollment in a PKI**

Configuring Custom Templates

---

Public Key Infrastructure Configuration Guide, Cisco IOS Release 15S

186
Configuration Examples for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI

Verifying the SDP Registrar Example

The following sample output from the show running-config command verifies that the certificate server “cs1” was configured and associated with the SDP exchange between the registrar and petitioner:

```
Router# show running-config
Building configuration...
Current configuration : 5902 bytes
!
! Last configuration change at 09:34:44 GMT Sat Jan 31 2004!

version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname pki-36a
!
boot-start-marker
boot-end-marker
!
logging buffered 32768 debugging
no logging console
!
enable secret 5 $1$b3jz$CKquLGjFIE3AdXA2/Rl9./
enable password lab
!
no clock timezone GMT 0

!
no ip subnet-zero
!
!
ip cef
!
ip domain name company.com
!
ip host msca-root
!
ip host yni-u10
!
ip host pki-36a 10.23.2.131
!
ip host pki-36a.company.com 10.23.2.131
!
!
crypto pki server cs1
issuer-name CN=company,L=city,C=US
hash sha1
lifetime crl 336
lifetime certificate 730
!
crypto pki trustpoint pki-36a
enrollment url http://pki-36a:80
ip-address FastEthernet0/0
revocation-check none
!
crypto pki trustpoint cs1
revocation-check crl
rsakeypair cs1 2048
!
!
crypto pki certificate chain pki-36a
certificate 03
308201D0 30820139 A0030201 02020103 300D0609 2A864886 F70D0101 04050030
```
public key infrastructure configuration guide, cisco ios release 15s

setting up secure device provisioning for enrollment in a PKI

Verifying the SDP Registrar Example

34110B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D033043 31333130
39553303 310F306700 30030502 33030309 33030303 3A030131 38530160 092A8648
86F700D1 09081309 310F3067 323E22E2 32301E06 092A8648 86F700D1 09021611
700B6920 33366126 63967963 6F2E636F 6D305C30 0600902A 846868F7 0D010101
05003448 00304802 4100AFFA 8F429618 112FAB9D 01F3352E 59DD3D2D AE67E31D
370AC4DA 619735DF 9C4F6A13 64A8B563 2C395CF0 1578BB73 07BD6E41 A18CA629
19181AB5 3B032E3C 4D031029 020001A5 300203B8 0B03035B 1008F043 03D025A0
301F0603 551D2302 18301660 141DA8B1 71652961 3F6D9F09 2F902A3C 2BADB137
C63000D6 092A8648 86F700D1 01040500 03811800 67BA1E41 327C3D31 D642CB39
AD5B7313 95868683 B950DF14 3B8C5B4A 3B4579C9 79128A9D 296922E9
4EEF2DF9 A7B54A12 AB1FC081 09951CE3 08BF6D9 9F81B90A E3A4AC56 D1049268
C924E88F 11CEB37F C750320C 03DFE6D4 C8B3E739 C8C76359 CE939A97 B51B3F7F
3FF9AD62 9CFDB6CF 829C5A14 36D0A236 A1CCFEAE
quit

certificate ca 01

38020241 30201AA A0030201 02020101 30000609 2A86486F 86F700D1 04050030
34110B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D033043 31333130
39553303 310F306700 30030502 33030309 33030303 3A030131 38530160 092A8648
86F700D1 09081309 310F3067 323E22E2 32301E06 092A8648 86F700D1 09021611
700B6920 33366126 63967963 6F2E636F 6D305C30 0600902A 846868F7 0D010101
05003448 00304802 4100AFFA 8F429618 112FAB9D 01F3352E 59DD3D2D AE67E31D
370AC4DA 619735DF 9C4F6A13 64A8B563 2C395CF0 1578BB73 07BD6E41 A18CA629
19181AB5 3B032E3C 4D031029 020001A5 300203B8 0B03035B 1008F043 03D025A0
301F0603 551D2302 18301660 141DA8B1 71652961 3F6D9F09 2F902A3C 2BADB137
C63000D6 092A8648 86F700D1 01040500 03811800 67BA1E41 327C3D31 D642CB39
AD5B7313 95868683 B950DF14 3B8C5B4A 3B4579C9 79128A9D 296922E9
4EEF2DF9 A7B54A12 AB1FC081 09951CE3 08BF6D9 9F81B90A E3A4AC56 D1049268
C924E88F 11CEB37F C750320C 03DFE6D4 C8B3E739 C8C76359 CE939A97 B51B3F7F
3FF9AD62 9CFDB6CF 829C5A14 36D0A236 A1CCFEAE
quit

certificate ca 01

crypto provisioning registrar
pki-server cs1

crypto isakmp policy 1
hash sha

crypto ipsec transform-set test_transformset esp-aes

crypto map test_cryptomap 10 ipsec-isakmp
set peer 10.23.1.10
define security-association lifetime seconds 1800

quit
set transform-set test_transformset
match address 170

interface Loopback0
  ip address 10.23.2.131 255.255.255.255
  no ip route-cache cef
  no ip route-cache
  no ip mroute-cache

interface FastEthernet0/0
  ip address 10.23.2.2 255.255.255.192
  no ip route-cache cef
  no ip route-cache
duplex auto
speed auto
crypto map test_cryptomap

interface FastEthernet1/0
  no ip address
  shutdown
duplex auto
  speed auto
  ip default-gateway 10.23.2.62
  ip http server
  no ip http secure-server
  ip classless
  ip route 0.0.0.0 0.0.0.0 10.23.2.62

  access-list 170 permit ip host 10.23.2.2 host 10.23.1.10
dialer-list 1 protocol ip permit

  control-plane

  line con 0
  exec-timeout 0 0
  speed 115200
  line aux 0
  line vty 0 4
  password lab
  login

end

Verifying the SDP Petitioner Example

After the SDP exchange is complete, the petitioner automatically enrolls with the registrar and obtain a certificate. The following sample output through the `show running-config` command shows the automatically generated configuration, which verifies that the trustpoint is really there:

```
Router# show running-config
Building configuration...
Current configuration : 4650 bytes

! Last configuration change at 09:34:53 GMT Sat Jan 31 2004
! version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
hostname pki-36b
```
Verifying the SDP Petitioner Example

Setting Up Secure Device Provisioning for Enrollment in a PKI

```
boot-start-marker
!
logging buffered 32768 debugging
no logging console
enable secret 5 $1$JYgw$060JXGl6dERLZpU9J3gb.
enable password lab
!
no crypto engine accelerator
!
!
!
Untitled-1
```
crypto isakmp policy 1
    hash sha
    !
crypto ipsec transform-set test_transformset esp-aes
    !
crypto map test_cryptomap 10 ipsec-isakmp
    set peer 10.23.2.2
    set security-association lifetime seconds 1800
    set transform-set test_transformset
    match address 170
    !
interface Ethernet0/0
    ip address 10.23.1.10 255.255.255.192
    no ip route-cache cef
    no ip route-cache
    no ip mroute-cache
    half-duplex
    crypto map test_cryptomap
    !
interface Ethernet0/1
    no ip address
    shutdown
    half-duplex
    !
interface Ethernet0/2
    no ip address
    shutdown
    half-duplex
    !
interface Ethernet0/3
    no ip address
    shutdown
    half-duplex
    !
interface Serial1/0
    no ip address
    shutdown
    serial restart-delay 0
    !
interface Serial1/1
    no ip address
    shutdown
    serial restart-delay 0
    !
interface Serial1/2
    no ip address
    shutdown
    serial restart-delay 0
    !
interface Serial1/3
    no ip address
    shutdown
    serial restart-delay 0
    !
ip default-gateway 10.23.1.62
ip http server
no ip http secure-server
ip classless
ip route 0.0.0.0 0.0.0.0 10.23.1.62
!
access-list 170 permit ip host 10.23.1.10 host 10.23.2.2
dialer-list 1 protocol ip permit
!
control-plane
!
line con 0
    exec-timeout 0 0
    speed 115200
Adding AAA Lists to a RADIUS or TACACS+ Server Examples

This section contains the following configuration examples:

**TACACS+ AAA Server Database Example**

In the following example, user information has been added to a TACACS+ AAA database. The username is “user1.” The password is “cisco.” Two Cisco IOS configuration template variables are configured for “user1”: $iosconfig1 and $iosconfig2. The variables replace $1 and $2 in the configuration template file. The subject name “CN=user1, O=company, C=US” is also configured. This subject name replaces the subject name field in the subsequent enrollment request (PKCS10) that is received from the petitioning device.

```
user = user1
password = clear "pswd"
service=tti
   ! The certificate server inserts the following subject name to the certificate.
   set subjectname="CN=user1, O=company, C=US"
   ! Up to nine template variables may be added.
   set iosconfig1="ntp server 10.3.0.1"
   set iosconfig2="hostname user1-vpn"
```

**RADIUS AAA Server Database Example**

User information has been added to the RADIUS AAA server database in the following example. The username is “user1.” The password is “cisco.” Two Cisco IOS configuration template variables are configured for “user1”: $iosconfig1 and $iosconfig2. The variables replace $1 and $2 in the configuration template file. The subject name “CN=user1, O=company, C=US” is also configured. This subject name replaces the subject name field in the subsequent enrollment request (PKCS10) that is received from the petitioning device.

```
user = user1
password = clear "pswd"
radius=company
reply_attributes=9,1="tti:subjectname=CN=user1, O=company, C=US"
   ! Up to nine template variables may be added.
   9,1="tti:iosconfig1=ntp server 10.3.0.5"
   9,1="tti:iosconfig2=hostname user1-vpn"
```

**AAA List on a TACACS+ and a RADIUS AAA Server Example**

The following is a configuration example showing that AAA authentication has been configured on a TACACS+ server and that AAA authorization has been configured on a RADIUS server.

```
Router(config)# tacacs-server host 10.0.0.48 key cisco
```

**Note**

Authentication and authorization usually point to the same server.
Using a Configuration Template File Example

You can use a different configuration template file on the basis of the introducer name. For example, if you have multiple template files for different users, each with the username in the filename, configure the following under the registrar:

```
Router(config)# crypto provisioning registrar
Router (tti-registrar)# pki-server cs1
Router (tti-registrar)# template config tftp://server/config-$n.txt
```

In this example, the default configuration file shown in the section “Default Template for the Configuration File, on page 163” is used because the `template config` command does not reference a CGI script.

CGI Script Example

The following example would execute a CGI script named "mysdpcgi":

```
Router(config)# crypto provisioning registrar
Router (tti-registrar)# pki-server cs1
Router (tti-registrar)# template config tftp://server/cgi-bin/mysdpcgi post
```

The following is an example CGI script, named "mysdpcgi", that would be executed with the example `template config` command above:

```
#!/usr/bin/perl -w
# for debugging use the -debug form
# use CGI (-debug);
use CGI;
# base64 decoding is being used.
use MIME::Base64;
# The following has been commented out, but left for your information.
#
# Reading everything that has been received from stdin and writing it to the debug log to
# see what has been sent from the registrar.
#
# Remember to reset the STDIN pointer so that the normal CGI processing can get the input.
# print STDERR "mysdpcgi.cgi dump of stdin:\n";
# if($ENV{'REQUEST_METHOD'} eq "GET"){
# $input_data = $ENV{'QUERY_STRING'};
# } else {
# $data_length = $ENV{'CONTENT_LENGTH'};
# $bytes_read = read(STDIN, $input_data, $data_length);
# }
# print STDERR $input_data,"\n";
# exit;

$query = new CGI;
my %av_table;
# A basic configuration file is being sent back, therefore it is being indicated as plain
# text in the command below.
print $query->header ("text/plain");
print "\n";
# For testing, parameters can be passed in so that the test applications can
# see what has been received.
# print STDERR "The following are the raw AV pairs mysdpcgi.cgi received:\n";
```
# for each $key ($query->param) {
# print STDERR "! $key is: \n";
# $value = $query->param($key);
# print STDERR "! "$value;
# print STDERR "! \n";
#}
# The post process AV pairs are identical to those in Cisco IOS and may be used to produce
# AV pair specific configurations as needed.
%av_table = &postprocessavpairs($query->param);

# Decoded values may be written out.
# WARNING: Some error logs cannot handle the amount of data and will freeze.
# print STDERR "The following are the decoded AV pairs mysdpcgi.cgi received:\n";
# now write the values out
# while ( ($a, $v) = each(%av_table) ) {
# print STDERR "$a = $v\n";
#}
# Identifying the AV pairs and specifying them in the config.
while ( ($a, $v) = each(%av_table) ) {
  if ($a eq "TTIIosRunningConfig") {
    $search = "hostname ";
    $begin = index($v, $search) + length($search);
    $end = index($v, "\n", $begin);
    $hostname = substr($v, $begin, $end - $begin);
  }
  if ($a eq "TTIIosVersion") {
    $search = "Version ";
    $begin = index($v, $search) + length($search);
    $end = index($v, "(", $begin);
    $version = substr($v, $begin, $end - $begin);
  }
  print <<END_CONFIG;
  !
  ! Config auto-generated by sdp.cgi
  ! This is for SDP testing only and is not a real config
  !
  "$t
  "$c
  cry pki trust Version-$version-$hostname
  ! NOTE: The last line of the config must be 'end' with a blank line after the end
  # statement.
  END_CONFIG
  
  # Emulate IOS tti_postprocessavpairs functionality
  sub postprocessavpairs {
    @attributes = @_; # Combine any AV pairs that were split apart
    $n = 0; #element index counter
    while ($attributes[$n]) {
      # see if we are at the start of a set
      if ($attributes[$n] =~ /_0/) {
        # determine base attribute name
        $a = (split /_0/, $attributes[$n])[0];
        # set initial (partial) value
        $v = $query->param($attributes[$n]);
        # loop and pull the rest of the matching
        # attribute's values into v (would be
        # faster if we stop at first non-match)
        $c = $n+1;
        while ($attributes[$c]) {
          if ($attributes[$c] =~ /m/SA/) {
            $v = $v.$query->param($attributes[$c]);
          } $c++;
        }
        # store in the av hash table
        $av_table{$a} = $v;
      } else {
Configuring the Petitioner and Registrar for Certificate-Based Authentication Example

The following examples shows how to configure a petitioner to use the certificate issued by the trustpoint named mytrust:

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# crypto provisioning petitioner

Router(tti-petitioner)# trustpoint signing mytrust

Router(tti-petitioner)# end

The following example shows how to configure a registrar to verify the petitioner-signing certificate and to perform authorization lookups:

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# crypto provisioning registrar

Router(tti-registrar)# authentication trustpoint mytrust

Router(tti-registrar)# authorization login certificate

Router(tti-registrar)# authorization username subjectname all

Router(tti-registrar)# end

Configuring an Administrative Introducer Using Authentication and Authorization Lists Example

The following example shows how to configure an administrative introducer with the authentication list “authen-tac” and the authorization list “author-tac”:

Router# configure terminal
Enter configuration commands, one per line. End with CTRL/Z.
Router(config)# crypto provisioning registrar
Router(tti-registrar)# administrator
authentication list authen-tac
Router(tti-registrar)# administrator
authorization list author-tac
Router(tti-registrar)# end

### Additional References

#### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
<tr>
<td>Certificate enrollment</td>
<td>“Configuring Certificate Enrollment for a PKI” module</td>
</tr>
<tr>
<td>Certificate server configuration</td>
<td>“Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment” module</td>
</tr>
<tr>
<td>PKI AAA integration concepts and configuration tasks</td>
<td>“Configuration Revocation and Authorization of Certificates in a PKI” module</td>
</tr>
<tr>
<td>PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples</td>
<td>Cisco IOS Security Command Reference</td>
</tr>
<tr>
<td>USB token configuration</td>
<td>“Storing PKI Credentials” chapter in the Cisco IOS Security Configuration Guide: Secure Connectivity</td>
</tr>
<tr>
<td>Recommended cryptographic algorithms</td>
<td>Next Generation Encryption</td>
</tr>
</tbody>
</table>

#### MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None.</td>
<td>To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to</td>
<td><a href="http://www.cisco.com/cisco/web/">http://www.cisco.com/cisco/web/</a></td>
</tr>
<tr>
<td>download documentation, software, and tools. Use these resources to install</td>
<td>support/index.html</td>
</tr>
<tr>
<td>and configure the software and to troubleshoot and resolve technical</td>
<td></td>
</tr>
<tr>
<td>issues with Cisco products and technologies. Access to most tools on the</td>
<td></td>
</tr>
<tr>
<td>Cisco Support and Documentation website requires a Cisco.com user ID and</td>
<td></td>
</tr>
<tr>
<td>password.</td>
<td></td>
</tr>
</tbody>
</table>

Feature Information for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 7: Feature Information for SDP in a PKI

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure Device Provisioning (SDP) Connect Template</td>
<td>12.4(20)T</td>
<td>This feature provides the ability to configure a device for Internet connectivity through a service provider. The following sections provide information about this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Prerequisites for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI, on page 142</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SDP Overview</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- How SDP Works</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Default Templates for SDP Transaction Web Pages</td>
</tr>
</tbody>
</table>
## Feature Information

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| USB Token and Secure Device Provisioning (SDP) Integration | 12.4(15)T | This feature provides the ability to provision remote devices using a USB token as a mechanism to transfer credentials from one network device to a remote device through SDP. The following sections provide information about this feature:  
- Prerequisites for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI, on page 142  
- SDP Leveraging USB Tokens, on page 151  
- Enabling the SDP Registrar for Certificate-Based Authorization  

The following commands were introduced: binary file, crypto key move rsa, template file.  

**Note** For other documentation on this topic, see the “Feature Information for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI, on page 197" section. |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| SDP Expanded Template CGI Support | 12.4(6)T | This feature allows users to configure the SDP registrar to send a bootstrap configuration to the SDP petitioner based on not only the device name, but also its current Cisco IOS version and current configuration. The following sections provide information about this feature:  
  • SDP Introduction Phase  
  • Custom Configuration and File Template Variable Expansion Rules, on page 158  
  • Default Template for the Configuration File, on page 163  
  • Enabling the SDP Registrar and Adding AAA Lists to the Server, on page 172  
  • CGI Script Example, on page 193  

The following command was modified by this feature: `template config`. |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Secure Device Provisioning (SDP) Start Page       | 12.4(4)T  | This feature allows users to configure their browsers to begin the TTI transaction by contacting the registrar’s introduction URL through a start page. Thus, users no longer have to begin the TTI transaction from the welcome page on the petitioner. The following sections provide information about this feature:  
  • How Custom Templates Work with SDP  
  • Configuring Custom Templates, on page 185 |
| Administrative Secure Device Provisioning Introducer | 12.3(14)T | This feature allows you to act as an administrative introducer to introduce a device into a PKI network and then provide a username as the device name for the record locator in the AAA database. The following sections provide information about this feature:  
  • Authentication and Authorization Lists for an Administrative Introducer, on page 155  
  • Configuring an Administrative Introducer, on page 182  
  
  The following commands were introduced by this feature:  
  administrator authentication list, administrator authorization list. |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Easy Secure Device Deployment | 12.3(8)T | This feature introduces support for SDP, which offers a web-based enrollment interface that enables network administrators to deploy new devices in large networks. The following sections provide information about this feature:  
  - Information About Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI, on page 143  
  - Enabling the SDP Registrar and Adding AAA Lists to the Server, on page 172  
  - The following commands were introduced or modified: `crypto wui tti petitioner`, `crypto wui tti registrar`, `pki-server`, `template config`, `template username`, `trustpoint (tti-petitioner)`. |

| Easy Secure Device Deployment AAA Integration | 12.3(8)T | This feature integrates an external AAA database, allowing the SDP introducer to be authenticated against a AAA database instead of having to use the enable password of the local Cisco certificate server. The following sections provide information about this feature:  
  - How SDP Uses an External AAA Database, on page 154  
  - Enabling the SDP Registrar and Adding AAA Lists to the Server, on page 172  
  - The following commands were introduced or modified: `authentication list (tti-registrar)`, `authorization list (tti-registrar)`, `debug crypto wui template config`, `template username`. |
### Feature Information for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure Device Provisioning (SDP) Certificate-Based Authorization</td>
<td>12.3(14)T</td>
<td>This feature allows certificates issued by other authority (CA) servers to be used for SDP introductions. The following sections provide information about this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Feature Information for Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI, on page 197</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Enabling the SDP Registrar for Certificate-Based Authorization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following commands were introduced by this feature: administrator authentication list, administrator authorization list</td>
</tr>
<tr>
<td>iPhone SDP</td>
<td>15.1(2)T</td>
<td>With the introduction of the Cisco IOS 15.1(2)T and Apple iPhone OS 3.0 releases, Apple iPhones are supported on Cisco IOS network devices. Cisco IOS routers use the SDP registrar to deploy iPhones so that network applications can be accessed securely through an IPSec VPN, SCEP server, and PKI certificate deployment technologies.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following sections provide information about this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following commands were introduced by this feature: match authentication trustpoint, match certificate, match url, mime-type, template location, template variable p, url-profile.</td>
</tr>
</tbody>
</table>

**Public Key Infrastructure Configuration Guide, Cisco IOS Release 15S**
CHAPTER 6

Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment

This module describes how to set up and manage a Cisco IOS certificate server for public key infrastructure (PKI) deployment. A certificate server embeds a simple certificate server, with limited certification authority (CA) functionality, into the Cisco IOS software. Thus, the following benefits are provided to the user:

- Easier PKI deployment by defining default behavior. The user interface is simpler because default behaviors are predefined. That is, you can leverage the scaling advantages of PKI without all of the certificate extensions that a CA provides, thereby allowing you to easily enable a basic PKI-secured network.
- Direct integration with Cisco IOS software.

Note

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

- Finding Feature Information, page 204
- Prerequisites for Configuring a Cisco IOS Certificate Server, page 204
- Restrictions for Configuring a Cisco IOS Certificate Server, page 205
- Information About Cisco IOS Certificate Servers, page 205
- How to Set Up and Deploy a Cisco IOS Certificate Server, page 213
- Configuration Examples for Using a Certificate Server, page 242
- Where to Go Next, page 252
- Additional References, page 252
- Feature Information for the Cisco IOS Certificate Server, page 253
Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring a Cisco IOS Certificate Server

Planning Your PKI Before Configuring the Certificate Server

Before configuring a Cisco IOS certificate server, it is important that you have planned for and chosen appropriate values for the settings you intend to use within your PKI (such as certificate lifetimes and certificate revocation list (CRL) lifetimes). After the settings have been configured in the certificate server and certificates have been granted, settings cannot be changed without having to reconfigure the certificate server and reenrolling the peers. For information on certificate server default settings and recommended settings, see the section "Certificate Server Default Values and Recommended Values, on page 229.”

Enabling an HTTP Server

The certificate server supports Simple Certificate Enrollment Protocol (SCEP) over HTTP. The HTTP server must be enabled on the router for the certificate server to use SCEP. (To enable the HTTP server, use the `ip http server` command.) The certificate server automatically enables or disables SCEP services after the HTTP server is enabled or disabled. If the HTTP server is not enabled, only manual PKCS10 enrollment is supported.

Note

To take advantage of automatic CA certificate and key pair rollover functionality for all types of certificate servers, Cisco IOS Release 12.4(4)T or a later release must be used and SCEP must be used as the enrollment method.

Configuring Reliable Time Services

Time services must be running on the router because the certificate server must have reliable time knowledge. If a hardware clock is unavailable, the certificate server depends on manually configured clock settings, such as Network Time Protocol (NTP). If there is not a hardware clock or the clock is invalid, the following message is displayed at bootup:

% Time has not been set. Cannot start the Certificate server.

After the clock has been set, the certificate server automatically switches to running status.

For information on manually configuring clock settings, see the section "Setting Time and Calendar Services" in the chapter "Performing Basic System Management" of the Cisco IOS Network Management Configuration Guide.
“crypto ca” to “crypto pki” CLI Change

As of Cisco IOS Release 12.3(7)T, all commands that begin as "crypto ca" have been changed to begin as "crypto pki." Although the router continues to accept crypto ca commands, all output is read back as crypto pki.

Restrictions for Configuring a Cisco IOS Certificate Server

The certificate server does not provide a mechanism for modifying the certificate request that is received from the client; that is, the certificate that is issued from the certificate server matches the requested certificate without modifications. If a specific certificate policy, such as name constraints, must be issued, the policy must be reflected in the certificate request.

Information About Cisco IOS Certificate Servers

RSA Key Pair and Certificate of the Certificate Server

The certificate server automatically generates a 1024-bit Rivest, Shamir, and Adelman (RSA) key pair. You must manually generate an RSA key pair if you prefer a different key pair modulus. For information on completing this task, see the section "Generating a Certificate Server RSA Key Pair, on page 213."

Note

The recommended modulus for a certificate server RSA key pair is 2048 bits.

The certificate server uses a regular Cisco IOS RSA key pair as its CA key. This key pair must have the same name as the certificate server. If you do not generate the key pair before the certificate server is created on the router, a general-purpose key pair is automatically generated during the configuration of the certificate server.

As of Cisco IOS Release 12.3(11)T and later releases, the CA certificate and CA key can be backed up automatically one time after they are generated by the certificate server. As a result, it is not necessary to generate an exportable CA key for backup purposes.

What to Do with Automatically Generated Key Pairs in Cisco IOS Software Prior to Release 12.3(11)T

If the key pair is automatically generated, it is not marked as exportable. Thus, you must manually generate the key pair as exportable if you want to back up the CA key. For information on how to complete this task, see the section "Generating a Certificate Server RSA Key Pair, on page 213."

How the CA Certificate and CA Key Are Automatically Archived

At initial certificate server setup, you can enable the CA certificate and the CA key to be automatically archived so that they may be restored later if either the original copy or the original configuration is lost.

When the certificate server is turned on the first time, the CA certificate and CA key is generated. If automatic archive is also enabled, the CA certificate and the CA key is exported (archived) to the server database. The archive can be in PKCS12 or privacy-enhanced mail (PEM) format.
This CA key backup file is extremely important and should be moved immediately to another secured place.

- This archiving action occurs only one time. Only the CA key that is (1) manually generated and marked exportable or (2) automatically generated by the certificate server is archived (this key is marked nonexportable).
- Autoarchiving does not occur if you generate the CA key manually and mark it “nonexportable.”
- In addition to the CA certificate and CA key archive file, you should also regularly back up the serial number file (.ser) and the CRL file (.crl). The serial file and the CRL file are both critical for CA operation if you need to restore your certificate server.
- It is not possible to manually back up a server that uses nonexportable RSA keys or manually generated, nonexportable RSA keys. Although automatically generated RSA keys are marked as nonexportable, they are automatically archived once.

**Certificate Server Database**

The Cisco IOS certificate server stores files for its own use and may publish files for other processes to use. Critical files generated by the certificate server that are needed for its ongoing operation are stored to only one location per file type for its exclusive use. The certificate server reads from and writes to these files. The critical certificate server files are the serial number file (.ser) and the CRL storage location file (.crl). Files that the certificate server writes to, but does not read from again, may be published and available for use by other processes. An example of a file that may be published is the issued certificates file (.crt).

Performance of your certificate server may be affected by the following factors, which should be considered when you choose storage options and publication options for your certificate server files.

- The storage or publish locations you choose may affect your certificate server performance. Reading from a network location takes more time than reading directly from a router’s local storage device.
- The number of files you choose to store or publish to a specific location may affect your certificate server performance. The local Cisco IOS file system may not always be suitable for a large number of files.
- The file types you choose to store or publish may affect your certificate server performance. Certain files, such as the .crl files, can become very large.

**Note**

It is recommended that you store .ser and .crl files to your local Cisco IOS file system and publish your .crt files to a remote file system.

**Certificate Server Database File Storage**

The certificate server allows the flexibility to store different critical file types to different storage locations depending on the database level set (see the database level command for more information). When choosing storage locations, consider the file security needed and server performance. For instance, serial number files
and archive files (.p12 or .pem) might have greater security restrictions than the issued certificates file storage location (.crt) or the name file storage location (.cnm).

The table below shows the critical certificate server file types by file extension that may be stored to a specific location.

**Table 8: Certificate Server Storage Critical File Types**

<table>
<thead>
<tr>
<th>File Extension</th>
<th>File Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>.ser</td>
<td>The main certificate server database file.</td>
</tr>
<tr>
<td>.crl</td>
<td>The CRL storage location.</td>
</tr>
<tr>
<td>.crt</td>
<td>The issued certificates storage location.</td>
</tr>
<tr>
<td>.cnm</td>
<td>The certificate name and expiration file storage location.</td>
</tr>
<tr>
<td>.p12</td>
<td>The certificate server certificate archive file location in PKCS12 format.</td>
</tr>
<tr>
<td>.pem</td>
<td>The certificate server certificate archive file location in PEM format.</td>
</tr>
</tbody>
</table>

Cisco IOS certificate server files may be stored to three levels of specificity:

- Default location, NVRAM
- Specified primary storage location for all critical files
- Specified storage location for specific critical file(s).

A more specific storage location setting overrides a more general storage location setting. For instance, if you have not specified any certificate server file storage locations, all certificate server files are stored to NVRAM. If you specify a storage location for the name file, only the name file is stored there; all other files continue to be stored to NVRAM. If you then specify a primary location, all files except the name file is now stored to this location, instead of NVRAM.

---

**Note**

You may specify either .p12 or .pem; you cannot specify both types of archive files.

### Certificate Server Database File Publication

A publish file is a copy of the original file and is available for other processes to use or for your use. If the certificate server fails to publish a file, it does cause the server to shut down. You may specify one publish location for the issued certificates file and name file and multiple publish locations for the CRL file. See the table below for files types available for publication. You may publish files regardless of the database level that is set.
Trustpoint of the Certificate Server

If the certificate server also has an automatically generated trustpoint of the same name, then the trustpoint stores the certificate of the certificate server. After the router detects that a trustpoint is being used to store the certificate of the certificate server, the trustpoint is locked so that it cannot be modified.

Before configuring the certificate server you can perform the following:

- Manually create and set up this trustpoint (using the `crypto pki trustpoint` command), which allows you to specify an alternative RSA key pair (using the `rsakeypair` command).
- Specify that the initial autoenrollment key pair is generated on a specific device, such as a configured and available USB token, using the `on` command.

The automatically generated trustpoint and the certificate server certificate are not available for the certificate server device identity. Thus, any command-line interface (CLI) (such as the `ip http secure-trustpoint` command) that is used to specify the CA trustpoint to obtain certificates and authenticate the connecting client’s certificate must point to an additional trustpoint configured on the certificate server device.

If the server is a root certificate server, it uses the RSA key pairs and several other attributes to generate a self-signed certificate. The associated CA certificate has the following key usage extensions—Digital Signature, Certificate Sign, and CRL Sign.

After the CA certificate is generated, attributes can be changed only if the certificate server is destroyed.

A certificate server trustpoint must not be automatically enrolled using the `auto-enroll` command. Initial enrollment of the certificate server must be initiated manually and ongoing automatic rollover functionality may be configured with the `auto-rollover` command. For more information on automatic rollover functionality, see the section "Automatic CA Certificate and Key Rollover, on page 211.”

---

**Table 9: Certificate Server Publish File Types**

<table>
<thead>
<tr>
<th>File Extension</th>
<th>File Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>.crl</td>
<td>The CRL publish location.</td>
</tr>
<tr>
<td>.crt</td>
<td>The issued certificates publish location.</td>
</tr>
<tr>
<td>.cnm</td>
<td>The certificate name and expiration file publish location.</td>
</tr>
</tbody>
</table>
Certificate Revocation Lists (CRLs)

By default, CRLs are issued once every 168 hours (1 calendar week). To specify a value other than the default value for issuing the CRL, execute the `lifetime crl` command. After the CRL is issued, it is written to the specified database location as `ca-label.crl`, where `ca-label` is the name of the certificate server.

CRLs can be distributed through SCEP, which is the default method, or a CRL distribution point (CDP), if configured and available. If you set up a CDP, use the `cdp-url` command to specify the CDP location. If the `cdp-url` command is not specified, the CDP certificate extension is not included in the certificates that are issued by the certificate server. If the CDP location is not specified, Cisco IOS PKI clients automatically request a CRL from the certificate server with a SCEP GetCRL message. The CA then returns the CRL in a SCEP CertRep message to the client. Because all SCEP messages are enveloped and signed PKCS#7 data, the SCEP retrieval of the CRL from the certificate server is costly and not highly scalable. In very large networks, an HTTP CDP provides better scalability and is recommended if you have many peer devices that check CRLs. You may specify the CDP location by a simple HTTP URL string for example,

```
cdp-url http://my-cdp.company.com/filename.crl
```

The certificate server supports only one CDP; thus, all certificates that are issued include the same CDP. If you have PKI clients that are not running Cisco IOS software and that do not support a SCEP GetCRL request and wish to use a CDP, you may set up an external server to distribute CRLs and configure the CDP to point to that server. Or, you can specify a non-SCEP request for the retrieval of the CRL from the certificate server by specifying the `cdp-url` command with the URL in the following format where `cs-addr` is the location of the certificate server:

```
cdp-url http://cs-addr/cgi-bin/pkiclient.exe?operation=GetCRL
```

It is the responsibility of the network administrator to ensure that the CRL is available from the location that is specified through the `cdp-url` command.

In order to force the parser to retain the embedded question mark within the specified location, enter Ctrl-v prior to the question mark. If this action is not taken, CRL retrieval through HTTP returns an error message.

The CDP location may be changed after the certificate server is running through the `cdp-url` command. New certificates contain the updated CDP location, but existing certificates are not reissued with the newly specified CDP location. When a new CRL is issued, the certificate server uses its current cached CRL to generate a new CRL. (When the certificate server is rebooted, it reloads the current CRL from the database.) A new CRL cannot be issued unless the current CRL has expired. After the current CRL expires, a new CRL is issued only after a certificate is revoked from the CLI.

Certificate Server Error Conditions

At startup, the certificate server checks the current configuration before issuing any certificates. It reports the last known error conditions through the `show crypto pki server` command output. Example errors can include any of the following conditions:

- Storage inaccessible
• Waiting for HTTP server
• Waiting for time setting

If the certificate server experiences a critical failure at any time, such as failing to publish a CRL, the certificate server automatically enters a disabled state. This state allows the network administrator to fix the condition; thereafter, the certificate server returns to the previous normal state.

Certificate Enrollment Using a Certificate Server

A certificate enrollment request functions as follows:

• The certificate server receives the enrollment request from an end user, and the following actions occur:
  • A request entry is created in the enrollment request database with the initial state. (See the table below for a complete list of certificate enrollment request states.)
  • The certificate server refers to the CLI configuration (or the default behavior any time a parameter is not specified) to determine the authorization of the request. Thereafter, the state of the enrollment request is updated in the enrollment request database.

• At each SCEP query for a response, the certificate server examines the current request and performs one of the following actions:
  • Responds to the end user with a "pending" or "denied" state.
  • Generates and signs the appropriate certificate and stores the certificate in the enrollment request database.

If the connection of the client has closed, the certificate server waits for the client to request another certificate. All enrollment requests transition through the certificate enrollment states that are defined in the table below. To see current enrollment requests, use the `crypto pki server request pkcs10` command.

<table>
<thead>
<tr>
<th>Certificate Enrollment State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>authorized</td>
<td>The certificate server has authorized the request.</td>
</tr>
<tr>
<td>denied</td>
<td>The certificate server has denied the request for policy reasons.</td>
</tr>
<tr>
<td>granted</td>
<td>The CA core has generated the appropriate certificate for the certificate request.</td>
</tr>
<tr>
<td>initial</td>
<td>The request has been created by the SCEP server.</td>
</tr>
<tr>
<td>malformed</td>
<td>The certificate server has determined that the request is invalid for cryptographic reasons.</td>
</tr>
<tr>
<td>pending</td>
<td>The enrollment request must be manually accepted by the network administrator.</td>
</tr>
</tbody>
</table>
SCEP Enrollment

All SCEP requests are treated as new certificate enrollment requests, even if the request specifies a duplicate subject name or public key pair as a previous certificate request.

Types of CA Servers Subordinate and Registration Authorities (RAs)

CA servers have the flexibility to be configured as a subordinate certificate server or an RA-mode certificate server.

Why Configure a Subordinate CA?

A subordinate certificate server provides all the same features as a root certificate server. The root RSA key pairs are extremely important in a PKI hierarchy, and it is often advantageous to keep them offline or archived. To support this requirement, PKI hierarchies allow for subordinate CAs that have been signed by the root authority. In this way, the root authority can be kept offline (except to issue occasional CRL updates), and the subordinate CA can be used during normal operation.

Why Configure an RA-Mode Certificate Server?

A Cisco IOS certificate server can be configured to run in RA mode. An RA offloads authentication and authorization responsibilities from a CA. When the RA receives a SCEP or manual enrollment request, the administrator can either reject or grant it on the basis of local policy. If the request is granted, it is forwarded to the issuing CA, and the CA automatically generates the certificate and return it to the RA. The client can later retrieve the granted certificate from the RA.

An RA is the authority charged with recording or verifying some or all of the data required for the CA to issue certificates. In many cases the CA undertakes all of the RA functions itself, but where a CA operates over a wide geographical area or when there is security concern over exposing the CA to direct network access, it may be administratively advisable to delegate some of the tasks to an RA and leave the CA to concentrate on its primary tasks of signing certificates and CRLs.

CA Server Compatibility

In Cisco IOS Release 15.1(2)T, new functionality was introduced that allows the IOS CA server in RA mode to interoperate with more than one type of CA server. See Configuring a Certificate Server to Run in RA Mode, on page 225 for more information.

Automatic CA Certificate and Key Rollover

CAs—root CAs, subordinate CAs, and RA-mode CAs—like their clients, have certificates and key pairs with expiration dates that need to be renewed when the current certificate and key pair are about to expire. When a root CA’s certificate and key pair are expiring it must generate a self-signed rollover certificate and key pair. If a subordinate CA or an RA-mode CA’s certificate and key pair are expiring, it requests a rollover certificate and key pair from its superior CA, obtaining the superior CA’s new self-signed rollover certificates at the same time. The CA must distribute the new CA rollover certificate and keys too all its peers. This process, called rollover, allows for continuous operation of the network while the CAs and their clients are switching from an expiring CA certificate and key pair to a new CA certificate and key pair.
Rollover relies on the PKI infrastructure requirements of trust relationships and synchronized clocks. The PKI trust relationships allow (1) the new CA certificate to be authenticated, and (2) the rollover to be accomplished automatically without the loss of security. Synchronized clocks allow the rollover to be coordinated throughout your network.

**Automatic CA Certificate Rollover How It Works**

The CA server must have rollover configured. All levels of CAs must be automatically enrolled and have auto-rollover enabled. CA clients support rollover automatically when automatically enrolled. For more information about clients and automatic rollover, see the section "Automatic Certificate Enrollment" in the chapter "Configuring Certificate Enrollment for a PKI".

After CAs have rollover enabled and their clients are automatically enrolled, there are three stages to the automatic CA certificate rollover process.

**Stage One: Active CA Certificate and Key Pair Only**

In stage one, there is an active CA certificate and key pair only.

**Stage Two: Rollover CA Certificate and Key Pair Generation and Distribution**

In stage two, the rollover CA certificate and key pair are generated and distributed. The superior CA generates a rollover certificate and key pair. After the CA successfully saves its active configuration, the CA is ready to respond to client requests for the rollover certificate and key pair. When the superior CA receives a request for the new CA certificate and key pair from a client, the CA responds by sending the new rollover CA certificate and key pair to the requesting client. The clients store the rollover CA certificate and key pair.

---

**Note**

When a CA generates its rollover certificate and key pair, it must be able to save its active configuration. If the current configuration has been altered, saving of the rollover certificate and key pair does not happen automatically. In this case, the administrator must save the configuration manually or rollover information is lost.

---

**Stage Three: Rollover CA Certificate and Key Pair Become the Active CA Certificate and Key Pair**

In stage three, the rollover CA certificate and key pair become the active CA certificate and key pair. All devices that have stored a valid rollover CA certificate rename the rollover certificate to the active certificate and the once-active certificate and key pair are deleted.

After the CA certificate rollover, you may observe the following deviation from usual certificate lifetime and renewal time:

- The lifetime of the certificates issued during rollover is lower than the preconfigured value.
- In specific conditions, the renew time may be inferior to the configured percentage of the actual lifetime. The difference observed can be of up to 20% in cases where the certificate lifetime is less than one hour.

These differences are normal, and result from jitter (random time fluctuation) introduced by the algorithm on the Certificate server. This task is performed to avoid the hosts participating to the PKI synchronize their enrollment timer, which could result in congestion on the Certificate Server.
The lifetime fluctuations that occur do not affect proper functionning of the PKI, since the differences always result in a shorter lifetime, thus remaining within maximum configured lifetime for certificates.

**Support for Specifying a Cryptographic Hash Function**

Secure Hash Algorithm (SHA) support allows a user to specify a cryptographic hash function for Cisco IOS certificate servers and clients. The cryptographic hash functions that can be specified are Message Digest algorithm 5 (MD5), SHA-1, SHA-256, SHA-384, or SHA-512.

Cisco no longer recommends using MD5; instead, you should use SHA-256 where supported. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

See the "Configuring a Subordinate Certificate Server" task for more information on specifying the `hash (ca-trustpoint)` and `hash (cs-server)` commands that are used to implement this feature.

**How to Set Up and Deploy a Cisco IOS Certificate Server**

**Generating a Certificate Server RSA Key Pair**

Perform this task to manually generate an RSA key pair for the certificate server. Manually generating a certificate server RSA key pair allows you to specify the type of key pair you want to generate, to create an exportable key pair for backup purposes, to specify the key pair storage location, or to specify the key generation location.

If you are running Cisco IOS Release 12.3(8)T or earlier releases, you may want to create an exportable certificate server key pair for backup, or archive purposes. If this task is not performed, the certificate server automatically generates a key pair, which is not marked as exportable. Automatic CA certificate archiving was introduced in Cisco IOS Release 12.3(11)T.

As of Cisco IOS Release 12.4(11)T and later releases, if your router has a USB token configured and available, the USB token can be used as cryptographic device in addition to a storage device. Using a USB token as a cryptographic device allows RSA operations such as key generation, signing, and authentication of credentials to be performed on a USB token. The private key never leaves the USB token and is not exportable. The public key is exportable. For titles of specific documents about configuring a USB token and making it available to use as a cryptographic device, see the "Related Documents" section.

It is recommended that the private key be kept in a secure location and that you regularly archive the certificate server database.
Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

### SUMMARY STEPS

1. enable
2. configure terminal
3. crypto key generate rsa [general-keys | usage-keys | signature | encryption] [label key-label] [exportable] [modulus modulus-size] [storage devicename:] [on devicename:]
4. crypto key export rsa key-label pem {terminal | url url | 3des | des} passphrase
5. crypto key import rsa key-label pem {usage-keys | signature | encryption} {terminal | url url} [exportable] [on devicename:] passphrase
6. exit
7. show crypto key mypubkey rsa

### DETAILED STEPS

<table>
<thead>
<tr>
<th></th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>crypto key generate rsa [general-keys</td>
<td>usage-keys</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>• The storage keyword specifies the key storage location.</td>
</tr>
<tr>
<td></td>
<td>Router (config)# crypto key generate rsa label mycs</td>
<td>• When specifying a label name by specifying the key-label argument, you must use the same name for the label that you plan to use for the certificate server (through the crypto pki server cs-label command). If a key-label argument is not specified, the default value, which is the fully qualified domain name (FQDN) of the router, is used.</td>
</tr>
<tr>
<td></td>
<td>exportable modulus 2048</td>
<td>If the exportable RSA key pair is manually generated after the CA certificate has been generated, and before issuing the no shutdown command, then use the crypto ca export pkcs12 command to export a PKCS12 file that contains the certificate server certificate and the private key.</td>
</tr>
</tbody>
</table>
Purpose

Command or Action | Purpose |
--- | --- |

- By default, the modulus size of a CA RSA key is 1024 bits. The recommended modulus for a CA RSA key is 2048 bits. The range for a modulus size of a CA RSA key is from 350 to 4096 bits.
- The **on** keyword specifies that the RSA key pair is created on the specified device, including a Universal Serial Bus (USB) token, local disk, or NVRAM. The name of the device is followed by a colon (:).

**Note**

Keys created on a USB token must be 2048 bits or less.

**Step 4**

**crypto key export rsa**  
**key-label**  
**pem**  
{**terminal** | **url**}  
{**3des** | **des**}  
**passphrase**

(Optional) Exports the generated RSA key pair.

Allows you to export the generated keys.

**Example:**

Router (config)# crypto key export rsa mycs pem url nvram: 3des PASSWORD

**Step 5**

**crypto key import rsa**  
**key-label**  
**pem**  
{**usage-keys** | **signature** | **encryption**}  
{**terminal** | **url**}  
{**exportable** | **on devicename:**}  
**passphrase**

(Optional) Imports RSA key pair.

To create the imported keys on a USB token, use the **on** keyword and specify the appropriate device location.

If you exported the RSA keys using the **exportable** keyword and you want to change the RSA key pair to non-exportable, import the key back to the certificate server without the **exportable** keyword. The key cannot be exported again.

**Example:**

Router (config)# crypto key import rsa mycs2 pem url nvram:mycs PASSWORD

**Step 6**

**exit**

Exits global configuration.

**Example:**

Router (config)# exit

**Step 7**

**show crypto key mypubkey rsa**

Displays the RSA public keys of your router.

**Example:**

Router# show crypto key mypubkey rsa

**Example**

The following example generates a general usage 1024-bit RSA key pair on a USB token with the label "ms2" with crypto engine debugging messages shown:

Router(config)# crypto key generate rsa on usbtoken0 label ms2 modulus 2048
The name for the keys will be: ms2
% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be on-token, non-exportable...
Jan 7 02:41:40.895: crypto_engine: Generate public/private keypair [OK]
Jan 7 02:44:09.623: crypto_engine: Create signature
Jan 7 02:44:10.467: crypto_engine: Verify signature
Jan 7 02:44:10.467: CryptoEngine0: CRYPTO_RSA_CREATE_PUBKEY(hw)(ipsec)
Jan 7 02:44:10.467: CryptoEngine0: CRYPTO_RSA_PUB_DECRYPT(hw)(ipsec)

Now, the on-token keys labeled "ms2" may be used for enrollment.

The following example shows the successful import of an encryption key to a configured and available USB tokens:

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
crypto key import rsa encryption on usbtoken0 url nvram:e password
% Importing public Encryption key or certificate PEM file...
filename [e-encr.pub]?
Reading file from nvram:e-encr.pub
% Importing private Encryption key PEM file...
Source filename [e-encr.prv]?
Reading file from nvram:e-encr.prv
% Key pair import succeeded.

Configuring Certificate Servers

Prerequisites for Automatic CA Certificate Rollover

When configuring a certificate server, for automatic CA certificate rollover to run successfully, the following prerequisites are applicable for your CA servers:

- You must be running Cisco IOS Release 12.4(2)T or a later release on your CA servers.
- Your CA server must be enabled and fully configured with a reliable time of day, an available key pair, a self-signed, valid CA certificate associated with the key pair, a CRL, an accessible storage device, and an active HTTP/SCEP server.
- CA clients must have successfully completed automatic enrollment and have autoenrollment enabled with the same certificate server.

Note

If you are running Cisco IOS 12.4(2)T or earlier releases, only your root CA supports automatic CA certificate rollover functionality. Cisco IOS 12.4(4)T or later releases support all CAs--root CAs, subordinate CAs, and RA-mode CAs.

Restrictions for Automatic CA Certificate Rollover

When configuring a certificate server, in order for automatic CA certificate rollover to run successfully, the following restrictions are applicable:

- SCEP must be used to support rollover. Any device that enrolls with the PKI using an alternative to SCEP as the certificate management protocol or mechanism (such as enrollment profiles, manual
enrollment, or TFTP enrollment) is not be able to take advantage of the rollover functionality provided by SCEP.

- If you have automatic archive configured on your network and the archive fails, rollover does not occur because the certificate server does not enter the rollover state, and the rollover certificate and key pair is not automatically saved.

**Configuring a Certificate Server**

Perform this task to configure a Cisco IOS certificate server and enable automatic rollover.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. ip http server
4. crypto pki server *cs-label*
5. no shutdown
6. auto-rollover [time-period]

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip http server</td>
<td>Enables the HTTP server on your system.</td>
</tr>
<tr>
<td>Example: Router(config)# ip http server</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> crypto pki server <em>cs-label</em></td>
<td>Defines a label for the certificate server and enters certificate server configuration mode.</td>
</tr>
<tr>
<td>Example: Router(config)# crypto pki server server-pki</td>
<td><strong>Note</strong> If you manually generated an RSA key pair, the <em>cs-label</em> argument must match the name of the key pair.</td>
</tr>
<tr>
<td><strong>Step 5</strong> no shutdown</td>
<td>(Optional) Enables the certificate server.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Only use this command at this point if you want to use the preconfigured default functionality. That is, do not issue this command just yet if you plan to change any of the default settings as shown in the task “Configuring Certificate Server Functionality.”</td>
</tr>
<tr>
<td>Router(cs-server)# no shutdown</td>
<td><strong>Note</strong></td>
</tr>
</tbody>
</table>

**Step 6**

<table>
<thead>
<tr>
<th>auto-rollover</th>
<th>(Optional) Enables the automated CA certificate rollover functionality.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>- time-period—default is 30 days.</td>
</tr>
<tr>
<td>Router(cs-server)# auto-rollover 90</td>
<td></td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to configure the certificate server "ms2" where ms2 is the label of a 2048-bit RSA key pair:

```
Router(config)# crypto pki server ms2
Router(cs-server)# no shutdown
% Once you start the server, you can no longer change some of
% the configuration.
Are you sure you want to do this? [yes/no]: yes
% Certificate Server enabled.
Router(cs-server)#
```

```
Router(config)# show crypto pki server ms2
Certificate Server ms2:
    Status: enabled, configured
    CA cert fingerprint: 5A856122 4051347F 55E8C246 866D0AC3
    Granting mode is: manual
    Last certificate issued serial number: 0x1

Current storage dir: nvram:
    Database Level: Complete - all issued certs written as <serialnum>.cer
```

The following example shows how to enable automated CA certificate rollover on the server ms2 with the `auto-rollover` command. The `show crypto pki server` command shows that the automatic rollover has been configured on the server mycs with an overlap period of 25 days.

```
Router(config)# crypto pki server ms2
Router(cs-server)# auto-rollover 25
Router(cs-server)# no shutdown
%Some server settings cannot be changed after CA certificate generation.
% Exporting Certificate Server signing certificate and keys...
% Certificate Server enabled.
Router(cs-server)#
Router# show crypto pki server ms2
Certificate Server ms2:
    Status:enabled
    Server's configuration is locked (enter "shut" to unlock it)
    Issuer name:CN=mycs
```
Configuring a Subordinate Certificate Server

Perform this task to configure a subordinate certificate server to grant all or certain SCEP or manual certificate requests and to enable automatic rollover.

Note

- You must be running Cisco IOS Release 12.3(14)T or a later release. (Versions prior to Cisco IOS software Release 12.3(14)T support only one certificate server and no hierarchy; that is, subordinate certificate servers are not supported.)

- The root certificate server should be a Cisco IOS certificate server.

Note

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. enrollment [mode] [retry period minutes] [retry count number] url url [pem]
5. hash {md5 | sha1 | sha256 | sha384 | sha512}
6. exit
7. crypto pki server cs-label
8. issuer name DN-string
9. mode sub-cs
10. auto-rollover [time-period]
11. grant auto rollover {ca-cert | ra-cert}
12. hash {md5 | sha1 | sha256 | sha384 | sha512}
13. no shutdown
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  * Enter your password if prompted. |
| **Example:** Router> enable |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:** Router# configure terminal |
| **Step 3** crypto pki trustpoint name | Declares the trustpoint that your subordinate certificate server should use and enters ca-trustpoint configuration mode. |
| **Example:** Router (config)# crypto pki trustpoint sub |
| **Step 4** enrollment [mode] [retry period minutes] [retry count number] url url [pem] | Specifies the following enrollment parameters of the CA:  
  * (Optional) The **mode** keyword specifies the registration authority (RA) mode, if your CA system provides an RA. By default, RA mode is disabled.  
  * (Optional) The **retry period** keyword and **minutes** argument specifies the period, in minutes, in which the router waits before sending the CA another certificate request. Valid values are from 1 to 60. The default is 1.  
  * (Optional) The **retry count** keyword and **number** argument specifies the number of times a router will resend a certificate request when it does not receive a response from the previous request. Valid values are from 1 to 100. The default is 10.  
  * The **url** argument is the URL of the CA to which your router should send certificate requests.  
  **Note** With the introduction of Cisco IOS Release 15.2(1)T, an IPv6 address can be added to the **http**: enrolment method. For example: http://[ipv6-address]:80. The IPv6 address must be enclosed in brackets in the URL. See the enrollment url (ca-trustpoint) command page for more information on the other enrollment methods that can be used.  
  * (Optional) The **pem** keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request. |
| **Example:** Router (ca-trustpoint)# enrollment url http://caserver.myexample.com  
  Router (ca-trustpoint)# enrollment url http://[2001:DB8:1:1::1]:80 |
### Command or Action | Purpose |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong> hash {md5</td>
<td>sha1</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router (ca-trustpoint)# hash sha384</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router (ca-trustpoint)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> crypto pki server cs-label</td>
<td>Enables a Cisco IOS certificate server and enters cs-server configuration mode. <strong>Note</strong> The subordinate server must have the same name as the trustpoint that was created in Step 3 above.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# crypto pki server sub</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> issuer name DN-string</td>
<td>(Optional) Specifies the DN as the CA issuer name for the certificate server.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(cs-server) # issuer-name CN=sub CA, O=Cisco, C=us</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> mode sub-cs</td>
<td>Places the PKI server into sub-certificate server mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(cs-server)# mode sub-cs</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> auto-rollover [time-period]</td>
<td>(Optional) Enables the automated CA certificate rollover functionality.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(cs-server)# auto-rollover 90</td>
<td>- time-period --default is 30 days.</td>
</tr>
</tbody>
</table>

**Step 11**

grant auto rollover {ca-cert | ra-cert}

**Example:**

Router(cs-server)# grant auto rollover ca-cert

(Optional) Automatically grants reenrollment requests for subordinate CAs and RA-mode CAs without operator intervention.

- **ca-cert** -- Specifies that the subordinate CA rollover certificate is automatically granted.
- **ra-cert** -- Specifies that the RA-mode CA rollover certificate is automatically granted.

**Note**

If this is the first time that a subordinate certificate server is enabled and enrolled, the certificate request must be manually granted.

**Step 12**

hash {md5 | sha1 | sha256 | sha384 | sha512}

**Example:**

Router(cs-server)# hash sha384

(Optional) Sets the hash function for the signature that the Cisco IOS certificate authority (CA) uses to sign all of the certificates issued by the server.

- **md5** — Specifies that MD5, the default hash function, is used. (No longer recommended).
- **sha1** — Specifies that the SHA-1 hash function is used. (No longer recommended).
- **sha256** — Specifies that the SHA-256 hash function is used.
- **sha384** — Specifies that the SHA-384 hash function is used.
- **sha512** — Specifies that the SHA-512 hash function is used.

**Step 13**

no shutdown

**Example:**

Router(cs-server)# no shutdown

Enables or reenables the certificate server.

If this is the first time that a subordinate certificate server is enabled, the certificate server generates the key and obtains its signing certificate from the root certificate server.

---

**Examples**

If the certificate server fails to enable or if the certificate server has trouble handling the request that has been configured, you can use the **debug crypto pki server** command to troubleshoot your configuration as shown in the following below (Clock Not Set and Trustpoint Not Configured). Here, "ms2" refers to the label of a 2048-bit RSA key pair.

```plaintext
Router# debug crypto pki server
```
Clock Not Set

Router(config)# crypto pki server ms2
Router(cs-server)# mode sub-cs
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key % or type Return to exit
Password:
Jan 6 20:57:37.667: CRYPTO_CS: enter FSM: input state initial, input signal no shut
Re-enter password:
Jan 6 20:57:45.303: CRYPTO_CS: starting enabling checks
% Jan 6 20:57:45.303: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]
% Time has not been set. Cannot start the Certificate server

Trustpoint Not Configured

Router(config)# crypto pki server ms2
Router(cs-server)# mode sub-cs
Router(cs-server)# no shutdown
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key % or type Return to exit
Password:
Jan 6 21:00:15.961: CRYPTO_CS: enter FSM: input state initial, input signal no shut.
Jan 6 21:00:15.961: CRYPTO_CS: enter FSM: input state initial, input signal time set.
Jan 6 21:03:34.313: CRYPTO_CS: exit FSM: new state initial.
Jan 6 21:03:34.313: CRYPTO_CS: cs config has been unlocked
Re-enter password:
Jan 6 21:03:44.413: CRYPTO_CS: starting enabling checks
Jan 6 21:03:44.413: CRYPTO_CS: associated trust point 'sub' does not exist; generated automatically
Jan 6 21:03:44.417: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]
Jan 6 21:04:03.993: CRYPTO_CS: nvram filesystem
Jan 6 21:04:04.077: CRYPTO_CS: serial number 0x1 written.
You must specify an enrollment URL for this CA before you can authenticate it.
% Failed to authenticate the Certificate Authority
If the certificate server fails to obtain its signing certificate from the root certificate server, you can use the debug crypto pki transactions command to troubleshoot your configuration as shown in the following example:

Router# debug crypto pki transactions
Jan 6 21:07:00.311: CRYPTO_CS: enter FSM: input state initial, input signal time set
Jan 6 21:07:00.311: CRYPTO_CS: exit FSM: new state initial
Jan 6 21:07:00.311: CRYPTO_CS: cs config has been unlocked no sh
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key % or type Return to exit
Password:
Jan 6 21:07:03.535: CRYPTO_CS: enter FSM: input state initial, input signal no shut
Re-enter password:
Jan 6 21:07:10.619: CRYPTO_CS: starting enabling checks
Jan 6 21:07:10.619: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]
Jan 6 21:07:20.535: %SSH-5-ENABLED: SSH 1.99 has been enabled
Jan 6 21:07:25.991: CRYPTO_CS: serial number 0x1 written.
Jan 6 21:07:27.863: CRYPTO_CS: authenticating the CA 'sub'
Jan 6 21:07:27.863: CRYPTO_PKI: sending CA Certificate Request:
GET /cgi-bin/pkiclient.exe?operation=GetCACert&message=sub HTTP/1.0
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Cisco PKI)
Jan 6 21:07:27.867: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:07:27.871: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6 Certificate has the following attributes:
Fingerprint MD5: 328ACC02 52B25DB8 22F8F104 B6055B5B
Fingerprint SHA1: 02FD799D DD40C7A8 61DC53AB 1B89A3EA 2A729EE2
% Do you accept this certificate? [yes/no]:
Jan 6 21:07:30.879: CRYPTO_PKI: http connection opened
Jan 6 21:07:30.903: CRYPTO_PKI: HTTP response header:
HTTP/1.1 200 OK
Date: Thu, 06 Jan 2005 21:07:30 GMT
Server: server-IOS
Content-Type: application/x-x509-ca-cert
Expires: Thu, 06 Jan 2005 21:07:30 GMT
Last-Modified: Thu, 06 Jan 2005 21:07:30 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Accept-Ranges: none
Content-Type indicates we have received a CA certificate.
Jan 6 21:07:30.907: CRYPTO_PKI: CA certificate received.
Jan 6 21:07:30.927: CRYPTO_PKI: crypto_pki_authenticate_tp_cert()
Jan 6 21:07:30.927: CRYPTO_PKI: trustpoint sub authentication status = 0 y Trustpoint CA certificate accepted.
% Certificate request sent to Certificate Authority
% Enrollment in progress...
Router (cs-server)#
Jan 6 21:07:51.772: CRYPTO_CA: certificate not found
Jan 6 21:07:51.772: CRYPTO_CA: certificate not found
Jan 6 21:07:52.460: CRYPTO_CS: Publishing 213 bytes to crl file nvram:sub.crl
Jan 6 21:07:52.348: CRYPTO_CS: enrolling the server's trustpoint 'sub'
Jan 6 21:07:54.352: CRYPTO_CS: exit FSM: new state check failed
Jan 6 21:07:54.356: CRYPTO_PKI: transaction PKCSReq completed
Jan 6 21:07:54.356: CRYPTO_PKI: status:
Jan 6 21:07:55.016: CRYPTO_PKI: Certificate Request Fingerprint MD5: 1BA027DB 1C7860C7 EC188F65 6435EC80
Jan 6 21:07:55.016: CRYPTO_PKI: Certificate Request Fingerprint SHA1: 840DB52C E17614CB 0C7BE187 DFC884D D32CAA75
Jan 6 21:07:56.508: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:07:56.516: CRYPTO_PKI: http connection opened
Jan 6 21:07:59.136: CRYPTO_PKI: received msg of 776 bytes
Jan 6 21:07:59.136: CRYPTO_PKI: HTTP response header: HTTP/1.1 200 OK
Jan 6 21:07:59.136: CRYPTO_PKI: HTTP response header:
Jan 6 21:07:59.136: CRYPTO_PKI: HTTP/1.1 200 OK
Jan 6 21:07:59.136: CRYPTO_PKI: HTTP response header: HTTP/1.1 200 OK
Jan 6 21:07:59.136: CRYPTO_PKI: HTTP response header: HTTP/1.1 200 OK
Date: Thu, 06 Jan 2005 21:07:57 GMT
Server: server-IOS
Content-Type: application/x-pki-message
Expires: Thu, 06 Jan 2005 21:07:57 GMT
Last-Modified: Thu, 06 Jan 2005 21:07:57 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Accept-Ranges: none
Jan 6 21:07:59.324: The PKCS #7 message has 1 verified signers.
Jan 6 21:07:59.324: signing cert: issuer-cn=root1
Jan 6 21:07:59.324: Signed Attributes:
Jan 6 21:08:00.788: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:08:00.788: CRYPTO_PKI: Using unresolved IP Address 192.0.2.6
Jan 6 21:08:00.796: CRYPTO_PKI: http connection opened
Jan 6 21:08:11.804: CRYPTO_PKI: received msg of 776 bytes
Jan 6 21:08:11.804: CRYPTO_PKI: HTTP response header: HTTP/1.1 200 OK
Date: Thu, 06 Jan 2005 21:08:01 GMT
Server: server-IOS
Content-Type: application/x-pki-message
Expires: Thu, 06 Jan 2005 21:08:01 GMT
Last-Modified: Thu, 06 Jan 2005 21:08:01 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Accept-Ranges: none
Jan 6 21:08:11.992: The PKCS #7 message has 1 verified signers.
Jan 6 21:08:11.992: signing cert: issuer-cn=root1
Jan 6 21:08:11.992: Signed Attributes:
Jan 6 21:08:21.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan 6 21:08:31.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan 6 21:08:41.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan 6 21:08:51.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan 6 21:09:01.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan 6 21:09:11.996: CRYPTO_PKI: resend GetCertInitial, 1
Jan 6 21:09:11.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan 6 21:09:11.996: CRYPTO_PKI: resend GetCertInitial for session: 0
Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment

Configuring Certificate Servers

If the certificate server fails to enable or if the certificate server has trouble handling the request that has been configured, you can use the `debug crypto pki server` command to troubleshoot the progress of an enrollment. This command can also be used to debug the root CA (turn it on at the root CA).

### Configuring a Certificate Server to Run in RA Mode

The Cisco IOS certificate server can act as an RA for a Cisco IOS CA or another third party CA. Read the details in Step 8 for more information about the `transparent` keyword option if a third-party CA is used.
SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. enrollment url url
5. subject-name x.500-name
6. exit
7. crypto pki server cs-label
8. mode ra [transparent]
9. auto-rollover [time-period]
10. grant auto rollover {ca-cert | ra-cert}
11. no shutdown
12. no shutdown

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>crypto pki trustpoint name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router (config)# crypto pki trustpoint ra-server</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Declares the trustpoint that your RA mode certificate server should use and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>enrollment url url</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router (ca-trustpoint)# enrollment url <a href="http://ca-server.company.com">http://ca-server.company.com</a></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Specifies the enrollment URL of the issuing CA certificate server (root certificate server).</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>subject-name x.500-name</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>(Optional) Specifies the subject name the RA uses.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Include &quot;cn=ioscs RA&quot; or &quot;ou=ioscs RA&quot; in the subject name so that the issuing CA certificate server can recognize the RA (see Step 7 below).</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Exits ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Enables a Cisco IOS certificate server and enters cs-server configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Places the PKI server into RA certificate server mode. Use the <code>transparent</code> keyword to allow the CA server in RA mode to interoperate with more than one type of CA server. When the <code>transparent</code> keyword is used, the original PKCS#10 enrollment message is not re-signed and is forwarded unchanged. This enrollment message makes the IOS RA certificate server work with CA servers like the Microsoft CA server.</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>(Optional) Enables the automatic CA certificate rollover functionality.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>(Optional) Automatically grants reenrollment requests for subordinate CAs and RA-mode CAs without operator intervention.</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>Enables the certificate server.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>After this command is issued, the RA automatically enrolls with the root certificate server. After the RA certificate has been successfully received, you must issue the <code>no shutdown</code> command again, which reenables the certificate server.</td>
</tr>
</tbody>
</table>
### Configuring the Root Certificate Server to Delegate Enrollment Tasks to the RA Mode Certificate Server

Perform the following steps on the router that is running the issuing certificate server; that is, configure the root certificate server that is delegating enrollment tasks to the RA mode certificate server.

#### Note
Granting enrollment requests for an RA is essentially the same process as granting enrollment requests for client devices--except that enrollment requests for an RA are displayed in the section "RA certificate requests" of the command output for the `crypto pki server info-requests` command.

#### SUMMARY STEPS

1. `enable`
2. `crypto pki server cs-label info requests`
3. `crypto pki server cs-label grant req-id`
4. `configure terminal`
5. `crypto pki server cs-label`
6. `grant ra-auto`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router&gt; enable</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 2</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>crypto pki server cs-label info requests</code></td>
<td>Displays the outstanding RA certificate request.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# crypto pki server root-server info requests</code></td>
<td></td>
</tr>
</tbody>
</table>

**Note** This command is issued on the router that is running the issuing certificate server.
### Command or Action | Purpose
--- | ---
**Step 3** | crypto pki server *cs-label* grant *req-id*<br>Example: Router# crypto pki server root-server grant 9 | Grants the pending RA certificate request. **Note** Because the issuing certificate server delegates the enrollment request verification task to the RA, you must pay extra attention to the RA certificate request before granting it.

| Step 4 | configure terminal<br>Example: Router# configure terminal | Enters global configuration mode.

| Step 5 | crypto pki server *cs-label*<br>Example: Router (config)# crypto pki server root-server | Enables a Cisco IOS certificate server and enters cs-server configuration mode.

| Step 6 | grant ra-auto<br>Example: Router(cs-server)# grant ra-auto | (Optional) Specifies that all enrollment requests from an RA are to be granted automatically. **Note** For the *grant ra-auto* command to work, you have to include "cn=ioscs RA" or "ou=ioscs RA" in the subject name of the RA certificate. (See Step 2 above.)

---

**What to Do Next**

After you have configured a certificate server, you can use the preconfigured default values or specify values through the CLI for the functionality of the certificate server. If you choose to specify values other than the defaults, see the following section, “Configuring Certificate Server Functionality, on page 229.”

**Configuring Certificate Server Functionality**

After you have enabled a certificate server and are in certificate server configuration mode, use any of the steps in this task to configure basic certificate server functionality values other than the default values.

**Certificate Server Default Values and Recommended Values**

The default values for a certificate server are intended to address a relatively small network (of about ten devices). For example, the database settings are minimal (through the *database level minimal* command) and the certificate server handles all CRL requests through SCEP. For larger networks, it is recommended that you use either the database setting “names” or “complete” (as described in the *database level* command) for possible audit and revocation purposes. Depending on the CRL checking policy, you should also use an external CDP in a larger network.
Certificate Server File Storage and Publication Locations

You have the flexibility to store file types to different storage and publication locations.

**SUMMARY STEPS**

1. `database url root-url`
2. `database url {cnm | crl | crt | p12 | pem | ser} root-url`
3. `database url {cnm | crl | crt} publish root-url`
4. `database level {minimal | names | complete}`
5. `database username username [password [encr-type] password]`
6. `database archive {pkcs12 | pem}[password encr-type] password ]`
7. `issuer-name DN-string`
8. `lifetime {ca-certificate | certificate} time`
9. `lifetime crl time`
10. `lifetime enrollment-request time`
11. `cdp-url url`
12. `no shutdown`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Specifies the primary location where database entries for the certificate server are written.</td>
</tr>
<tr>
<td><code>database url root-url</code></td>
<td>If this command is not specified, all database entries are written to NVRAM.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router (cs-server)#</td>
<td>database url tftp://cert-svr-db.company.com</td>
</tr>
</tbody>
</table>

| **Step 2** | Specifies certificate server critical file storage location by file type. |
| `database url {cnm | crl | crt | p12 | pem | ser} root-url` | Note |
| Example: | If this command is not specified, all critical files are stored to the primary location if specified. If the primary location is not specified, all critical files are stored to NVRAM. |
| Router (cs-server)# | database url ser nvram: |

<p>| <strong>Step 3</strong> | Specifies certificate server publish location by file type. |
| <code>database url {cnm | crl | crt} publish root-url</code> | Note |
| Example: | If this command is not specified, all publish files are stored to the primary location if specified. If the primary location is not specified, all publish files are stored to NVRAM. |
| Router (cs-server)# | database url crl publish tftp://csdb_specific_crl_files.company.com |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td>Controls what type of data is stored in the certificate enrollment database.</td>
</tr>
<tr>
<td>`database level {minimal</td>
<td>names</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router (cs-server)# database level complete</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The <code>complete</code> keyword produces a large amount of information; if it is issued, you should also specify an external TFTP server in which to store the data through the <code>database url</code> command.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>(Optional) Sets a username and password when a user is required to access a primary certificate enrollment database storage location.</td>
</tr>
<tr>
<td><code>database username username [password [encr-type] password]</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router (cs-server)# database username user password PASSWORD</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) Sets the CA key and CA certificate archive format and password to encrypt the file.</td>
</tr>
<tr>
<td>`database archive {pkcs12</td>
<td>pem}[password encr-type] password ]`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router (cs-server)# database archive pem</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>It is recommended that you remove the password from the configuration after the archive is finished.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>(Optional) Sets the CA issuer name to the specified distinguished name (DN-string). The default value is as follows: <code>issuer-name cn={cs-label}</code>.</td>
</tr>
<tr>
<td><code>issuer-name DN-string</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router (cs-server)# issuer-name my-server</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>(Optional) Specifies the lifetime, in days, of a CA certificate or a certificate. Valid values range from 1 day to 1825 days. The default CA certificate lifetime is 3 years; the default certificate lifetime is 1 year. The maximum certificate lifetime is 1 month less than the lifetime of the CA certificate.</td>
</tr>
<tr>
<td>`lifetime {ca-certificate</td>
<td>certificate} time`</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 9</th>
<th>lifetime crl time</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router (cs-server)# lifetime crl 333</td>
<td>(Optional) Defines the lifetime, in hours, of the CRL that is used by the certificate server. Maximum lifetime value is 336 hours (2 weeks). The default value is 168 hours (1 week).</td>
</tr>
<tr>
<td>Step 10</td>
<td>lifetime enrollment-request time</td>
<td>(Optional) Specifies how long an enrollment request should stay in the enrollment database before being removed. Maximum lifetime is 1000 hours.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router (cs-server)# lifetime enrollment-request 888</td>
<td></td>
</tr>
</tbody>
</table>

### Command or Action

<table>
<thead>
<tr>
<th>Step 11</th>
<th>cdp-url url</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Example:        | Router (cs-server)# cdp-url http://my-cdp.company.com | (Optional) Defines the CDP location to be used in the certificates that are issued by the certificate server. \ 
|                 |                                           | • The URL must be an HTTP URL.                                                                                                       |
|                 |                                           | If you have PKI clients that are not running Cisco IOS software and that do not support a SCEP GetCRL request, use the following URL format: |
|                 |                                           | http://server.company.com/certEnroll/filename.crl                                                                                 |
|                 |                                           | Or, if your Cisco IOS certificate server is also configured as your CDP, use the following URL format                                |
|                 |                                           | http://cs-addr/cgi-bin/pkiclient.exe?operation=GetCRL                                                                             |
|                 |                                           | where cs-addr is the location of the certificate server.                                                                             |
|                 |                                           | In order to force the parser to retain the embedded question mark within the specified location, enter Ctrl-v prior to the question mark. If this action is not taken, CRL retrieval through HTTP returns an error message. |
|                 |                                           | **Note** Although this command is optional, it is strongly recommended for any deployment scenario. | |
| Step 12         | no shutdown                               | Enables the certificate server. You should issue this command only after you have completely configured your certificate server.    |
| Example:        | Router (cs-server)# no shutdown           |                                                                                                                                 |

### Examples

The following example shows how to configure a CDP location where the PKI clients do not support SCEP GetCRL requests:

```
Router(config)# crypto pki server aaa
Router(cs-server)# database level minimum
```
Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment

Working with Automatic CA Certificate Rollover

Starting Automated CA Certificate Rollover Immediately

Use this task to initiate the automated CA certificate rollover process immediately on your root CA server.

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki server *cs-label* rollover cancel ]]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| **Step 3**        | crypto pki server *cs-label* rollover cancel ]]
| **Example:**      | Router(config)# crypto pki server mycs rollover |
|                   | Immediately starts the CA certificate rollover process by generating a shadow CA certificate.  |
|                   | To delete the CA certificate rollover certificate and keys, use the cancel keyword. |
Requesting a Certificate Server Client Rollover Certificate

Use this task to request a certificate server client’s rollover certificate.

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki server cs-label rollover request pkcs10 terminal

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto pki server cs-label rollover request pkcs10 terminal</td>
<td>Requests a client rollover certificate from the server.</td>
</tr>
<tr>
<td>Example: Router(config)# crypto pki server mycs rollover request pkcs10 terminal</td>
<td></td>
</tr>
</tbody>
</table>

Example

The following example shows a rollover certificate request being inputted into the server:

```
Router# crypto pki server mycs rollover request pkcs10 terminal
% Enter Base64 encoded or PEM formatted PKCS10 enrollment request.
% End with a blank line or "quit" on a line by itself.
-----BEGIN CERTIFICATE REQUEST-----
MIIBUTCBuwIBADASMRAYwMDAGCSqGSIb3DQEBAQADCg8GCSqGSIb3DQEBCwUAMIGFMA0GCSqGSIb3DQEBAQDA
vMIGFMA0GCSqGSIb3DQEBAQDA+WwIBADASMRAYwMDAGCSqGSIb3DQEBAQADCg8GCSqGSIb3DQEBCwUAMIGF
-----END CERTIFICATE REQUEST-----
```

Public Key Infrastructure Configuration Guide, Cisco IOS Release 15S
Exporting a CA Rollover Certificate

Use this task to export a CA rollover certificate.

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki export trustpoint pem {terminal | url url} [rollover]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>crypto pki export trustpoint pem {terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# crypto pki export mycs pem terminal rollover</td>
</tr>
<tr>
<td></td>
<td>Exports a CA shadow certificate.</td>
</tr>
</tbody>
</table>

Maintaining Verifying and Troubleshooting the Certificate Server Certificates and the CA

Managing the Enrollment Request Database

SCEP supports two client authentication mechanisms--manual and preshared key. Manual enrollment requires the administrator at the CA server to specifically authorize the enrollment requests; enrollment using preshared keys allows the administrator to preauthorize enrollment requests by generating a one-time password (OTP).

Use any of the optional steps within this task to help manage the enrollment request database by performing functions such as specifying enrollment processing parameters that are to be used by SCEP and by controlling the run-time behavior or the certificate server.
SUMMARY STEPS

1. `enable`
2. `crypto pki server cs-label grant all req-id`
3. `crypto pki server cs-label reject {all req-id}
4. `crypto pki server cs-label password generate minutes`
5. `crypto pki server cs-label revoke certificate-serial-number`
6. `crypto pki server cs-label request pkcs10 {url | terminal} [base64 | pem]
7. `crypto pki server cs-label info crl`
8. `crypto pki server cs-label info requests`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> crypto pki server cs-label grant all req-id</td>
<td>Grants all or specific SCEP requests.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# crypto pki server mycs grant all</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto pki server cs-label reject {all req-id}</td>
<td>Rejects all or specific SCEP requests.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Router# crypto pki server mycs reject all</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> crypto pki server cs-label password generate minutes</td>
<td>Generates a OTP for SCEP requests.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# crypto pki server mycs password generate 75</td>
<td>• <em>minutes</em> --Length of time, in minutes, that the password is valid. Valid values range from 1 to 1440 minutes. The default is 60 minutes.</td>
</tr>
<tr>
<td><strong>Note</strong> Only one OTP is valid at a time; if a second OTP is generated, the previous OTP is no longer valid.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> crypto pki server cs-label revoke certificate-serial-number</td>
<td>Revokes a certificate on the basis of its serial number.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• <em>certificate-serial-number</em> --One of the following options:</td>
</tr>
</tbody>
</table>
### Command or Action

| Step 6 | `crypto pki server cs-label request pkcs10 {url | terminal} {base64 | pem}` |
|--------|----------------------------------------------------------------------------|

**Example:**

```
Router# crypto pki server mycs request pkcs10 terminal pem
```

- **Purpose:**
  - Manually adds either a base64-encoded or PEM-formatted PKCS10 certificate enrollment request to the request database.
  - After the certificate is granted, it is displayed on the console terminal using base64 encoding.
  - **`pem`** -- Specifies the certificate that is returned with PEM headers automatically added to the certificate after the certificate is granted, regardless of whether PEM headers were used in the request.
  - **`base64`** -- Specifies the certificate that is returned without privacy-enhanced mail (PEM) headers, regardless of whether PEM headers were used in the request.

### Step 7

```
crypto pki server cs-label info crl
```

**Example:**

```
Router# crypto pki server mycs info crl
```

**Purpose:** Displays information regarding the status of the current CRL.

### Step 8

```
crypto pki server cs-label info requests
```

**Example:**

```
Router# crypto pki server mycs info requests
```

**Purpose:** Displays all outstanding certificate enrollment requests.

---

**Removing Requests from the Enrollment Request Database**

After the certificate server receives an enrollment request, the server can leave the request in a pending state, reject it, or grant it. The request stays in the enrollment request database for 1 week until the client polls the certificate server for the result of the request. If the client exits and never polls the certificate server, you can remove either individual requests or all requests from the database.

Use this task to remove requests from the database and allow the server to be returned to a clean slate with respect to the keys and transaction IDs. Also, you can use this task to help troubleshoot a SCEP client that may not be behaving properly.
SUMMARY STEPS

1. enable
2. crypto pki server cs-label remove \{all | req-id\}

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
- Enter your password if prompted. |
| **Example:** Router> enable |
| **Step 2** crypto pki server cs-label remove \{all | req-id\} | Removes enrollment requests from the enrollment request database. |
| **Example:** Router# crypto pki server mycs remove 15 |

Deleting a Certificate Server

Users can delete a certificate server from the PKI configuration if they no longer want it on the configuration. Typically, a subordinate certificate server or an RA is being deleted. However, users may delete a root certificate server if they are moving it to another device through the archived RSA keys.

Perform this task to delete a certificate server from your PKI configuration.

Note

When a certificate server is deleted, the associated trustpoint and key are also deleted.

SUMMARY STEPS

1. enable
2. configure terminal
3. no crypto pki server cs-label

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router &gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

#### Step 2

**configure terminal**

| Example:          | Enters global configuration mode. |
| Router            |         |
| # configure terminal |   |

#### Step 3

**no crypto pki server** *cs-label*

| Example:          | Deletes a certificate server and associated trustpoint and key. |
| Router (config)# no crypto pki server mycs | |

---

### Verifying and Troubleshooting Certificate Server and CA Status

Use any of the following optional steps to verify the status of the certificate server or the CA.

#### SUMMARY STEPS

1. **enable**
2. **debug crypto pki server**
3. **dir filesystem:**

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2**        | Enables debugging for a crypto PKI certificate server. |
| debug crypto pki server | • This command can be used for monitoring the progress of an enrollment and for troubleshooting if the certificate server fails to respond or if the certificate server has trouble handling the request that has been configured. |
| Example:           |         |
| Router# debug crypto pki server | |

| **Step 3**        | Displays a list of files on a file system. |
| dir filesystem :  |         |
Purpose

Command or Action | Purpose
--- | ---
Example: Router# dir slot0: | • This command can be used to verify the certificate server autoarchived file if the `database url` command was entered to point to a local file system. You should be able to at least see “cs-label.ser” and “cs-label.crl” files in the database.

Verifying CA Certificate Information

To obtain information relating to the CA certificates including the certificate server rollover process, rollover certificates, and timers, you may use any of the following commands.

| Note | These commands are not exclusive to shadow certificate information. If no shadow certificate exists, the following commands display the active certificate information.

SUMMARY STEPS

1. The `crypto pki certificate chain` command can be used to view the certificate chain details and to distinguish the current active certificate from the rollover certificate in the certificate chain. The following example shows a certificate chain with an active CA certificate and a shadow, or rollover, certificate:

2. The `crypto pki server info requests` command displays all outstanding certificate enrollment requests. The following example shows the output for shadow PKI certificate information requests:

3. The `show crypto pki certificates` command displays information about your certificate, the certification authority certificate, shadow certificates, and any registration authority certificates. The following example displays the certificate of the router and the certificate of the CA. There is no shadow certificate available. A single, general-purpose RSA key pair was previously generated, and a certificate was requested but not received for that key pair. Note that the certificate status of the router shows "Pending." After the router receives its certificate from the CA, the Status field changes to "Available" in the `show` output.

4. The `show crypto pki server` command displays the current state and configuration of the certificate server. The following example shows that the certificate server “routercs” has rollover configured. The CA auto-rollover time has occurred and the rollover, or shadow, PKI certificate is available. The status shows the rollover certificate fingerprint and rollover CA certificate expiration timer information.

5. The `show crypto pki trustpoints` command displays the trustpoints that are configured in the router. The following output shows that a shadow CA certificate is available and shows the SCEP capabilities reported during the last enrollment operation:

DETAILED STEPS

Step 1 The `crypto pki certificate chain` command can be used to view the certificate chain details and to distinguish the current active certificate from the rollover certificate in the certificate chain. The following example shows a certificate chain with an active CA certificate and a shadow, or rollover, certificate:
Example:

Router(config)# crypto pki certificate chain mica
certificate 06
certificate ca 01
! This is the peer's shadow PKI certificate.
certificate rollover 0B
! This is the CA shadow PKI certificate
certificate rollover ca 0A

Step 2  The crypto pki server info requests command displays all outstanding certificate enrollment requests. The following example shows the output for shadow PKI certificate information requests:

Example:

Router# crypto pki server myca info requests
Enrollment Request Database:
RA certificate requests:
ReqID State Fingerprint SubjectName
------------------------------------------------------------
RA rollover certificate requests:
ReqID State Fingerprint SubjectName
------------------------------------------------------------
Router certificates requests:
ReqID State Fingerprint SubjectName
------------------------------------------------------------
1 pending A426AF07FE3A4BB69062E0E47198E5BF hostname=client
Router rollover certificates requests:
ReqID State Fingerprint SubjectName
------------------------------------------------------------
2 pending B69062E0E47198E5BFA426AF07FE3A4E hostname=client

Step 3  The show crypto pki certificates command displays information about your certificate, the certification authority certificate, shadow certificates, and any registration authority certificates. The following example displays the certificate of the router and the certificate of the CA. There is no shadow certificate available. A single, general-purpose RSA key pair was previously generated, and a certificate was requested but not received for that key pair. Note that the certificate status of the router shows "Pending." After the router receives its certificate from the CA, the Status field changes to "Available" in the show output.

Example:

Router# show crypto pki certificates
Certificate
Subject Name
Name: myrouter.example.com
IP Address: 192.0.2.1
Serial Number: 04806682
Status: Pending
Key Usage: General Purpose
Fingerprint: 428125BD A3419600 3F6C7831 6CD8FA95 00000000
CA Certificate
Status: Available
Certificate Serial Number: 3051DF7123BEE31B8341DFE4B3A338E5F
Key Usage: Not Set

Step 4  The show crypto pki server command displays the current state and configuration of the certificate server. The following example shows that the certificate server "routercs" has rollover configured. The CA auto-rollover time has occurred and the rollover, or shadow, PKI certificate is available. The status shows the rollover certificate fingerprint and rollover CA certificate expiration timer information.
Example:

Router# show crypto pki server

Certificate Server routercs:
  Status: enabled, configured
  Issuer name: CN=walnutcs
  CA cert fingerprint: 800F5944 74337E5B C2DF6C52 9A7B1BDB
  Granting mode is: auto
  Last certificate issued serial number: 0x7
  CRL NextUpdate timer: 21:50:56 GMT Mar 5 2004
  Current storage dir: nvram:
  Database Level: Minimum - no cert data written to storage
  Rollover status: available for rollover
    Rollover CA cert fingerprint: 6AAF5944 74227A5B 23DF3E52 9A7F1FEF
    Rollover CA certificate expiration timer: 22:10:29 GMT Jan 29 2017

Step 5

The `show crypto pki trustpoints` command displays the trustpoints that are configured in the router. The following output shows that a shadow CA certificate is available and shows the SCEP capabilities reported during the last enrollment operation:

Example:

Router# show crypto pki trustpoints

Trustpoint vpn:
  Subject Name: cn=Cisco SSL CA
  o=Cisco Systems
  Serial Number: 0FEBBDC1B6F6D9D0EA7875875E4C695
  Certificate configured.
  Rollover certificate configured.
  Enrollment Protocol:
    SCEPv1, PKI Rollover

---

Configuration Examples for Using a Certificate Server

Configuring Specific Storage and Publication Locations Examples

The following example shows the configuration of a minimal local file system, so that the certificate server can respond quickly to certificate requests. The .ser and .crl files are stored on the local Cisco IOS file system for fast access, and a copy of all of the .crt files are published to a remote location for long-term logging.

```
crypto pki server myserver
  !Pick your database level.
  database level minimum
  !Specify a location for the .crt files that is different than the default local
  !Cisco IOS file system.
  database url crt publish http://url username user1 password secret
```

Note

Free space on the local file system should be monitored, in case the .crl file becomes too large.
The following example shows the configuration of a primary storage location for critical files, a specific storage location for the critical file serial number file, the main certificate server database file, and a password protected file publication location for the CRL file:

Router(config)# crypto pki server mycs
Router(cs-server)# database url ftp://cs-db.company.com

! % Server database url was changed. You need to move the existing database to the new location.
!
Router(cs-server)# database url ser nvram:

Router(cs-server)# database url crl publish ftp://crl.company.com username myname password

Router(cs-server)# end

The following output displays the specified primary storage location and critical file storage locations specified:

Router# show
Sep 3 20:19:34.216: %SYS-5-CONFIG_I: Configured from console by user on console
Router# show crypto pki server
Certificate Server mycs:
  Status: disabled
  Server's configuration is unlocked (enter "no shut" to lock it)
  Issuer name: CN=mycs
  CA cert fingerprint: -Not found-
  Granting mode is: manual
  Last certificate issued serial number: 0x0
  CA certificate expiration timer: 00:00:00 GMT Jan 1 1970
  CRL not present.
  Current primary storage dir: ftp://cs-db.company.com
  Current storage dir for .ser files: nvram:

  Database Level: Minimum - no cert data written to storage

The following output displays all storage and publication locations. The serial number file (.ser) is stored in NVRAM. The CRL file will be published to ftp://crl.company.com with a username and password. All other critical files will be stored to the primary location, ftp://cs-db.company.com.

Router# show running-config

Removing Enrollment Requests from the Enrollment Request Database Examples

The following examples show both the enrollment requests that are currently in the enrollment request database and the result after one of the enrollment requests has been removed from the database.

Enrollment Request Currently in the Enrollment Request Database

The following example shows that the `crypto pki server info requests` command has been used to display the enrollment requests that are currently in the Enrollment Request Database:

Router# crypto pki server myserver info requests
Enrollment Request Database:
RA certificate requests:
ReqID  State  Fingerprint  SubjectName
-----------------------------------------------
Router certificates requests:
crypto pki server remove Command Used to Remove One Enrollment Request

The following example shows that the `crypto pki server remove` command has been used to remove Enrollment Request 1:

```
Router# crypto pki server myserver remove 1
```

Enrollment Request Database After the Removal of One Enrollment Request

The following example shows the result of the removal of Enrollment Request 1 from the Enrollment Request Database:

```
Router# crypto pki server mycs info requests
Enrollment Request Database:
RA certificate requests:  
ReqID State Fingerprint SubjectName
-------------------------------------------------------------
2 pending 1B07F3021DAAB0F19F35DA25D01D8567 hostname=host1.company.com
1 denied 5322459D2DC70B3F8EF3D03A795CF636 hostname=host2.company.com
```

Autoarchiving the Certificate Server Root Keys Examples

The following output configurations and examples show what you might see if the `database archive` command has not been configured (that is, configured using the default value); if the `database archive` command has been configured to set the CA certificate and CA key archive format as PEM, without configuring a password; and if the `database archive` command has been configured to set the CA certificate and CA key archive format as PKCS12, with a password configured. The last example is sample content of a PEM-formatted archive file. The following example, "ms2" refers to the label of a 2048-bit key pair.

```
database archive Command Not Configured

The default is PKCS12, and the prompt for the password appears after the `no shutdown` command has been issued.
```

```
Router (config)# crypto pki server ms2
Router (cs-server)# no shutdown
% Ready to generate the CA certificate.
% Some server settings cannot be changed after CA certificate generation.
Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
% Note the next two lines, which are asking for a password.
% Please enter a passphrase to protect the private key.
Password:
% Certificate Server enabled.
Router (cs-server)# end
Router# dir nvram:
Directory of nvram:/
 125 -rw- 1693 <no date> startup-config
 126 ---- 5 <no date> private-config
    1 -rw- 32 <no date> myserver.ser
```

```
database archive Command and pem Keyword Configured

The prompt for the password appears after the `no shutdown` command has been issued.

```bash
Router (config)# crypto pki server ms2
Router (cs-server)# database archive pem
Router (cs-server)# no shutdown
% Ready to generate the CA certificate.
% Some server settings cannot be changed after CA certificate generation.
Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
% Note the next two lines, which are asking for a password.
% Please enter a passphrase to protect the private key.
Password:
% Certificate Server enabled.
Router (cs-server)# end
Router# dir nvram
Directory of nvram:/
  125 -rw- 1693 <no date> startup-config
  126 ----  5 <no date> private-config
  1 -rw-  32 <no date> myserver.ser
  2 -rw- 214 <no date> myserver.crl
! Note the next line showing that the format is PEM.
  3 -rw- 1705 <no date> myserver.pem
```

database archive Command and pkcs12 Keyword (and Password) Configured

When the password is entered, it is encrypted. However, it is recommended that you remove the password from the configuration after the archive has finished.

```bash
Router (config)# crypto pki server ms2
Router (cs-server)# database archive pkcs12 password cisco123
Router (cs-server)# no shutdown
% Ready to generate the CA certificate.
% Some server settings cannot be changed after CA certificate generation.
Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
! Note that you are not being prompted for a password.
% Certificate Server enabled.
Router (cs-server)# end
Router# dir nvram:
Directory of nvram:/
  125 -rw- 1693 <no date> startup-config
  126 ----  5 <no date> private-config
  1 -rw-  32 <no date> myserver.ser
  2 -rw- 214 <no date> myserver.crl
! Note that the next line indicates that the format is PKCS12.
  3 -rw- 1499 <no date> myserver.p12
```

PEM-Formatted Archive

The following sample output shows that autoarchiving has been configured in PEM file format. The archive consists of the CA certificate and the CA private key. To restore the certificate server using the backup, you would have to import the PEM-formatted CA certificate and CA key individually.
In addition to the CA certificate and CA key archive files, you should also back up the serial file (.ser) and the CRL file (.crl) regularly. The serial file and the CRL file are both critical for CA operation if you need to restore your certificate server.

Note

Router# more nvram:mycs.pem

-----BEGIN CERTIFICATE-----
MIIB9zCCAWCgAwIBAgIBATANBgkqhkiG9w0BAQQFADAPMQ0wCwYDVQQDEwRteWNz
MB4XDTE0MDEyNzYyNTQwNzEwMjIwX50LzAIeGQ7A0EHa5KgVpY8yQ5yTPReP8c3
BzNzMBgkghklG1w90OBQFEPACGBoAzgYwK0gYAEAllzkpXfnGDJHMpKgS1k71Ds
n23a1M29k5oo/QT8ve88mupjycZ2t8A2voiCuYm5178Zw5pMgjiJaZELB
BtuVvlnsEOofhCu/uOJvxfgS5xpshoz/F5J3xdg5Z2WWuIDAUYu9+QbS5ufG4
ZtBiPbAMGTFP452M0BCAWEAAhNaJMGEm6D31Vr07AQH/BAUAWKB/0zAbgBvQ8Q8B
AfSEBACAMWyhwYVDr0jBhBw8FoA8u/0/cUKlxzYzSxhVtb06vUJbdjEw9bYDRV0O
BBYEFs03viu5A/EMsBnFDW901CW686BhMA6CGSGqG5Ihol3DEEBBAA46GBAKlMoe2
4NoKEXMCXG1iobK20HzkKfF1/vpK0+q92rTnMUFhXLoqI8Wp5Cc7heace
0Tv2zcUAOa4rzx3rc2USiixKokWQ1MLuMsMN/sLTeHitO5uj/8GcrgiK20MAW6
ymf7?+TmB1SiJlWzstoXCr2hnLsJIMq/KffaD
-----END CERTIFICATE-----

The private key is protected by the password that is configured in "database archive pem password pwd" or that is entered when you are prompted for the password.

-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,006F8A52

Restoring a Certificate Server from Certificate Server Backup Files Examples

The following examples show that restoration is from a PKCS12 archive and that the database URL is NVRAM (the default).

Router# copy tftp://192.0.2.71/pkcs12=ser nvram:mycs.ser Destination filename [mycs.ser]?
32 bytes copied in 1.320 secs (24 bytes/sec)

Router# copy tftp://192.0.2.71/pkcs12=crl nvram:mycs.crl Destination filename [mycs.crl]?
214 bytes copied in 1.324 secs (162 bytes/sec)

Router# configure terminal
Router (config)# crypto pki import mycs pkcs12 tftp://192.0.2.71/pkcs12=ser cisco123 Source filename [backup.p12]?
CRYPTO_PKI: Imported PKCS12 file successfully.

Router (config)# crypto pki server mycs
I fill in any certificate server configuration here
Router (cs-server)# no shutdown % Certificate Server enabled.
Router (cs-server)# end
Router# show crypto pki server
Certificate Server mycs:
Status: enabled
Server's current state: enabled
Issuer name: CN=mycs
Last certificate issued serial number: 0x1
Current storage dir: nvram:

The following examples show that restoration is from a PEM archive and that the database URL is flash:

Router# copy tftp://192.0.2.71/backup.ser flash:mycs.ser
Destination filename [mycs.ser]?
32 bytes copied in 1.320 secs (24 bytes/sec)

Router# copy tftp://192.0.2.71/backup.crl flash:mycs.crl
Destination filename [mycs.crl]?
214 bytes copied in 1.324 secs (162 bytes/sec)

Router# configure terminal

% Because CA cert has Digital Signature usage, you need to import using the "usage-keys" keyword

Router (config)# crypto ca import mycs pem usage-keys terminal cisco123
% Enter PEM-formatted CA certificate.
% End with a blank line or "quit" on a line by itself.

-----BEGIN CERTIFICATE-----
MIIB9zCCACAgIBAgIBATANBgkqhkiG9w0BAQQAFAAPMQowCwYDVQQDEwRteWNzMB0gCzAwNDA1NzIzMTAwODAwNzNkMjIyMzAwMDAwMTAwMTAwMTAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAW
Subordinate Certificate Server Example

The following configuration and output is typical of what you might see after configuring a subordinate certificate server. Please be aware that "ms2" refers to a 2048-bit RSA key that was generated in an earlier step.

```
% Enter PEM-formatted encrypted private ENCRYPTION key.
% End with "quit" on a line by itself.
! Because the CA cert only has Digital Signature usage, skip the encryption part.
quilt
% PEM files import succeeded.
Router (config)# crypto pki server mycs
Router (cs-server)# database url flash:
! Fill in any certificate server configuration here.
Router (cs-server)# no shutdown
% Certificate Server enabled.
Router (cs-server)# end
Router #
show crypto pki server
Certificate Server mycs:
Status: enabled
Server's current state: enabled
Issuer name: CN=mycs
CA cert fingerprint: F04C2B75 E0243FBC 19806219 B1D77412
Granting mode is: manual
Last certificate issued serial number: 0x2
CA certificate expiration timer: 21:02:55 GMT Sep 2 2007
CRL NextUpdate timer: 21:02:58 GMT Sep 9 2004
Current storage dir: flash:
Database Level: Minimum - no cert data written to storage
```

Subordinate Certificate Server Example

The following configuration and output is typical of what you might see after configuring a subordinate certificate server. Please be aware that "ms2" refers to a 2048-bit RSA key that was generated in an earlier step.

```
Router (config)# crypto pki trustpoint sub
Router (ca-trustpoint)# enrollment url http://192.0.2.6
Router (ca-trustpoint)# rsa keypair ms2 2048
Router (ca-trustpoint)# exit
Router (config)# crypto pki server sub
Router (cs-server)# mode sub-cs
Router (ca-server)# no shutdown
% Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key
% or type Return to exit
Password: 
Jan 6 22:32:22.698: CRYPTO_CS: enter FSM: input state initial, input signal no shut
Re-enter password: 
Jan 6 22:32:30.302: CRYPTO_CS: starting enabling checks
Jan 6 22:32:30.306: CRYPTO_CS: key 'sub' does not exist; generated automatically [OK]
Jan 6 22:32:39.810: %SSH-5-ENABLED: SSH 1.99 has been enabled
Certificate has the following attributes:
Fingerprint MD5: 328ACC02 52B25DB8 22F8F104 B6055B5B
Fingerprint SHA1: 02FD799D DD40C7A8 61DC53AB 1E89A3EA 2A729EEE2
% Do you accept this certificate? [yes/no]:
Jan 6 22:32:44.830: CRYPTO_CS: nvram filesystem
Jan 6 22:32:44.922: CRYPTO_CS: serial number 0x1 written.
Jan 6 22:33:46.798: CRYPTO_CS: authenticating the CA 'sub'
Trustpoint CA certificate accepted.%
% Certificate request sent to Certificate Authority
% Enrollment in progress...
```

Subordinate Certificate Server Example

The following configuration and output is typical of what you might see after configuring a subordinate certificate server. Please be aware that "ms2" refers to a 2048-bit RSA key that was generated in an earlier step.

```
Router (config)# crypto pki trustpoint sub
Router (ca-trustpoint)# enrollment url http://192.0.2.6
Router (ca-trustpoint)# rsa keypair ms2 2048
Router (ca-trustpoint)# exit
Router (config)# crypto pki server sub
Router (cs-server)# mode sub-cs
Router (ca-server)# no shutdown
% Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key
% or type Return to exit
Password: 
Jan 6 22:32:22.698: CRYPTO_CS: enter FSM: input state initial, input signal no shut
Re-enter password: 
Jan 6 22:32:30.302: CRYPTO_CS: starting enabling checks
Jan 6 22:32:30.306: CRYPTO_CS: key 'sub' does not exist; generated automatically [OK]
Jan 6 22:32:39.810: %SSH-5-ENABLED: SSH 1.99 has been enabled
Certificate has the following attributes:
Fingerprint MD5: 328ACC02 52B25DB8 22F8F104 B6055B5B
Fingerprint SHA1: 02FD799D DD40C7A8 61DC53AB 1E89A3EA 2A729EEE2
% Do you accept this certificate? [yes/no]:
Jan 6 22:32:44.830: CRYPTO_CS: nvram filesystem
Jan 6 22:32:44.922: CRYPTO_CS: serial number 0x1 written.
Jan 6 22:33:46.798: CRYPTO_CS: authenticating the CA 'sub'
Trustpoint CA certificate accepted.%
% Certificate request sent to Certificate Authority
% Enrollment in progress...
```
Root Certificate Server Differentiation Example

When issuing certificates, the root certificate server (or parent subordinate certificate server) differentiates the certificate request from "Sub CA," "RA," and peer requests, as shown in the following sample output:

```
Router# crypto pki server server1 info req
Enrollment Request Database:
RA certificate requests:
ReqID State Fingerprint SubjectName
-----------------------------
Subordinate CS certificate requests:
ReqID State Fingerprint SubjectName
1 pending CB9977AD8A73B146D3221749999B0F66 hostname=host-subcs.company.com
```
**RA Mode Certificate Server Example**

The following output is typical of what you might see after having configured an RA mode certificate server:

```
Router-ra (config)# crypto pki trustpoint myra
Router-ra (ca-trustpoint)# enrollment url http://192.0.2.17
  Include "cn=ioscs RA" or "ou=ioscs RA" in the subject-name.
Router-ra (ca-trustpoint)# subject-name cn=myra, ou=ioscs RA, o=company, c=us
Router-ra (ca-trustpoint)# exit
Router-ra (config)# crypto pki server myra
Router-ra (cs-server)# mode ra
% Generating 1024 bit RSA keys ...[OK]
Certificate has the following attributes:
  Fingerprint MD5: 32661452 0DDA3CE5 8723B469 09AB9E85
  Fingerprint SHA1: 9785BBCD 6C67D27C C950E8D0 718C7A14 C0FE9C38
% Do you accept this certificate? [yes/no]: yes
  Trustpoint CA certificate accepted.
% Ready to request the CA certificate.
% Some server settings cannot be changed after the CA certificate has been requested.
Are you sure you want to do this? [yes/no]: yes
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
  password to the CA administrator in order to revoke your certificate.
  For security reasons your password will not be saved in the configuration.
  Please make a note of it.
Password:
Re-enter password:
  % The subject name in the certificate will include: cn=myra, ou=ioscs RA, o=company, c=us
% The subject name in the certificate will include: Router-ra.company.com
% Include an IP address in the subject name? [yes/no]: no
% Request certificate from CA? [yes/no]: yes
  Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
  The 'show crypto pki certificate' command will also show the fingerprint.
% Enrollment in progress...
Router-ra (cs-server)#
Sep 15 22:32:40.197: CRYPTO_PKI: Certificate Request Fingerprint MD5: 82B41A76 AF4EC87D AF093CD 07747D3A
Sep 15 22:32:40.201: CRYPTO_PKI: Certificate Request Fingerprint SHA1: 897CDF40 C6563EAA 0FED05F7 0115FD3A 4FFC5231
Sep 15 22:34:00.366: %PKI-6-CERTRET: Certificate received from Certificate Authority
Router-ra (cs-server)#
Router-ra (cs-server)# end
Router-ra(config)# show crypto pki server
```

The following output shows the enrollment request database of the issuing certificate server after the RA has been enabled:

```
Certificate Server myra:
  Status: enabled
  Issuer name: CN=myra
  CA cert fingerprint: 32661452 0DDA3CE5 8723B469 09AB9E85
  Note that the certificate server is running in RA mode
  Server configured in RA mode
  RA cert fingerprint: C65F5724 0E63B3CC BE7AE016 BE0D34FE
  Granting mode is: manual
  Current storage dir: nvram:
  Database Level: Minimum - no cert data written to storage
```

The following output shows the enrollment request database of the issuing certificate server after the RA has been enabled:
The RA certificate request is recognized by the issuing certificate server because "ou=ioscs RA" is listed in the subject name.

```
Router-ca# crypto pki server mycs info request
Enrollment Request Database:
Subordinate CA certificate requests:
ReqID State Fingerprint SubjectName
--------------------------------------------------------------
! The request is identified as RA certificate request.
RA certificate requests:
ReqID State Fingerprint SubjectName
--------------------------------------------------------------
12 pending 88F547A407FA0C90F97CDE8900A30CB0
hostname=Router-ra.company.com,cn=myra,ou=ioscs RA,o=company,c=us
Router certificates requests:
ReqID State Fingerprint SubjectName
--------------------------------------------------------------
! Issue the RA certificate.
Router-ca# crypto pki server mycs grant 12
```

The following output shows that the issuing certificate server is configured to issue a certificate automatically if the request comes from an RA:

```
Router-ca(config)# crypto pki server myra
Router-ca (cs-server)# grant ra-auto
% This will cause all certificate requests already authorized by known RAs to be automatically granted.
Are you sure you want to do this? [yes/no]: yes
Router-ca (cs-server)# end
Router-ca# show crypto pki server
Certificate Server mycs:
  Status: enabled
  Server's current state: enabled
  Issuer name: CN=mycs
  CA cert fingerprint: 32661452 0DDA3CE5 8723B469 09AB9E85
  ! Note that the certificate server will issue certificate for requests from the RA.
  Granting mode is: auto for RA-authorized requests, manual otherwise
  Last certificate issued serial number: 0x2
  Current storage dir: nvram:
  Database Level: Minimum - no cert data written to storage
```

The following example shows the configuration of "myra", an RA server, configured to support automatic rollover from "myca", the CA. After the RA server is configured, automatic granting of certificate reenrollment requests is enabled:

```
crypto pki trustpoint myra
  enrollment url http://myca
  subject-name ou=ioscs RA
  rsakeypair myra
crypto pki server myra
  mode ra
  auto-rollover
crypto pki server mycs
  grant auto rollover ra-cert
  auto-rollover 25
```
Enabling CA Certificate Rollover to Start Immediately Example

The following example shows how to enable automated CA certificate rollover on the server mycs with the `crypto pki server` command. The `show crypto pki server` command then shows the current state of the mycs server and that the rollover certificate is currently available for rollover.

```
Router(config)# crypto pki server mycs rollover
Jun 20 23:51:21.211:%PKI-4-NOSHADOWAUTOSAVE:Configuration was modified. Issue "write memory" to save new IOS CA certificate!
The config has not been automatically saved because the config has been changed.
Router# show crypto pki server
Certificate Server mycs:
  Status:enabled
  Server's configuration is locked (enter "shut" to unlock it)
  Issuer name:CN=mycs
  CA cert fingerprint:E7A5FABA 5D7AA26C F2A9F7B3 03CE229A
  Granting mode is:manual
  Last certificate issued serial number:0x2
  CA certificate expiration timer:00:49:26 PDT Jun 20 2008
  CRL NextUpdate timer:00:49:29 PDT Jun 28 2005
  Current storage dir:nvram:
  Database Level:Minimum - no cert data written to storage
  Rollover status:available for rollover
  ! Rollover certificate is available for rollover.
  Rollover CA certificate fingerprint:9BD7A443 00A6DD74 E4D9ED5F B7931BE0
  Rollover CA certificate expiration time:00:49:26 PDT Jun 20 2011
  Auto-Rollover configured, overlap period 25 days
```

Where to Go Next

After the certificate server is successfully running, you can either begin enrolling clients through manual mechanisms (as explained in the module "Configuring Certificate Enrollment for a PKI") or begin configuring SDP, which is a web-based enrollment interface, (as explained in the module "Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI.")

Additional References

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
<tr>
<td>PKI and security commands</td>
<td>Cisco IOS Security Command Reference</td>
</tr>
</tbody>
</table>
## Feature Information for the Cisco IOS Certificate Server

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting up and logging into a USB token</td>
<td>“Storing PKI Credentials” module in the Cisco IOS Security Configuration Guide: Secure Connectivity.</td>
</tr>
<tr>
<td>Recommended cryptographic algorithms</td>
<td>Next Generation Encryption</td>
</tr>
</tbody>
</table>

## Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>
### Feature Information for the Cisco IOS Certificate Server

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Cisco IOS USB Token PKI Enhancements-- Phase 2 | 12.4(11)T | This feature enhances USB token functionality by using the USB token as a cryptographic device. USB tokens may be used for RSA operations such as key generation, signing, and authentication. The following sections in this document provide information about this feature:  
- RSA Key Pair and Certificate of the Certificate Server, on page 205  
- Trustpoint of the Certificate Server  
- Generating a Certificate Server RSA Key Pair, on page 213 |
<p>| IOS Certificate Server (CS) Split Database | 12.4(4)T | This feature allows the user to set storage locations and publish locations for specific certificate server file types. The following sections provide information about this feature: The following command was modified by this feature: <code>database url</code> |</p>
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subordinate/RA Mode IOS Certificate Server (CS) Rollover</td>
<td>12.4(4)T</td>
<td>This feature expands on Certificate Authority (CA) Key Rollover introduced in 12.4(2)T to allow CA certificate rollover for subordinate CAs and RA-mode CAs. This functionality allows the rollover expiring CA certificates and keys and to have these changes propagate through the PKI network without manual intervention. The following sections provide information about this feature: The following command was modified by this feature: <code>grant auto rollover</code></td>
</tr>
<tr>
<td>Certificate Authority (CA) Key Rollover</td>
<td>12.4(2)T</td>
<td>This feature introduces the ability for root or subordinate CAs to roll over expiring CA certificates and keys and to have these changes propagate through the PKI network without manual intervention. The following sections provide information about this feature: The following commands were introduced or modified by this feature: <code>auto-rollover</code>, <code>crypto pki certificate chain</code>, <code>crypto pki export pem</code>, <code>crypto pki server info request</code>, <code>crypto pki server</code>, <code>show crypto pki certificates</code>, <code>show crypto pki server</code>, and <code>show crypto pki trustpoint</code></td>
</tr>
<tr>
<td>Cisco IOS Certificate Server</td>
<td>12.3(8)T</td>
<td>This feature introduces support for the Cisco IOS certificate server, which offers users a CA that is directly integrated with Cisco IOS software to more easily deploy basic PKI networks. The following sections provide information about this feature:</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>The Certificate Server Auto Archive Enhancement</td>
<td>12.3(11)T</td>
<td>This enhancement enables the CA certificate and CA key to be backed up automatically just once after they are generated by the certificate server. As a result, it is not necessary to generate an exportable CA key if CA backup is desirable. The following sections provide information about this feature: The following commands were introduced by this feature: <code>crypto pki server remote, database archive</code></td>
</tr>
<tr>
<td>The Certificate Server Registration Authority (RA) Mode enhancement</td>
<td>12.3(7)T</td>
<td>A certificate server can be configured to run in RA mode. The following section provides information about this feature: The following commands were introduced by this feature: <code>grant ra-auto, lifetime enrollment-requests</code></td>
</tr>
<tr>
<td>PKI Status 1</td>
<td>12.3(11)T</td>
<td>This enhancement provides a quick snapshot of current trustpoint status. The following section provides information about this enhancement: The following command was modified by this enhancement: <code>show crypto pki trustpoints</code></td>
</tr>
<tr>
<td>Subordinate Certificate Server 1</td>
<td>12.3(14)T</td>
<td>This enhancement allows you to configure a subordinate certificate server to grant all or certain SCEP or manual certificate requests. The following section provides information about this enhancement: The following command was introduced by this enhancement: <code>mode sub-cs</code></td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RSA 4096-bit Key Generation in Software Crypto Engine Support</td>
<td>15.1(1)T</td>
<td>The range value for the <em>modulus</em> keyword value for the <code>crypto key generate rsa</code> command is extended from 360 to 2048 bits to 360 to 4096 bits.</td>
</tr>
<tr>
<td>IOS PKI Server RA Mode Support for Non-IOS CA Servers</td>
<td>15.1(2)T</td>
<td>This enhancement allows the IOS CA server in RA mode to interoperate with more than one type of CA server. The following section provides information about this feature: The <em>transparent</em> keyword was added to the <code>mode ra</code> command to allow the CA server in RA mode to interoperate with more than one type of CA server.</td>
</tr>
<tr>
<td>Public Key Infrastructure (PKI) IPv6 Support for VPN Solutions</td>
<td>15.2(1)T</td>
<td>The <code>enrollment url (ca-trustpoint)</code> command was modified to allow the specification of an IPv6 address in the URL for the CA.</td>
</tr>
</tbody>
</table>

1 This is a minor enhancement. Minor enhancements are not typically listed in Feature Navigator.
Storing PKI Credentials

Public key infrastructure (PKI) credentials, such as Rivest, Shamir, and Adelman (RSA) keys and certificates can be stored in a specific location on the router, such as NVRAM and flash memory or on a USB eTtoken 64 KB smart card. USB tokens provide secure configuration distribution, RSA operations such as on-token key generation, signing, and authentication, and the storage of Virtual Private Network (VPN) credentials for deployment.

- Finding Feature Information, page 259
- Prerequisites for Storing PKI Credentials, page 260
- Restrictions for Storing PKI Credentials, page 260
- Information About Storing PKI Credentials, page 261
- How to Configure PKI Storage, page 263
- Configuration Examples for PKI Storage, page 277
- Additional References, page 278
- Feature Information for Storing PKI Credentials, page 279

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.
Prerequisites for Storing PKI Credentials

Prerequisites for Specifying a Local Certificate Storage Location
Before you can specify the local certificate storage location, your system should meet the following requirements:

- A Cisco IOS Release 12.4(2)T PKI-enabled image or a later image
- A platform that supports storing PKI credentials as separate files
- A configuration that contains at least one certificate
- An accessible local file system

Prerequisites for Specifying USB Token Storage for PKI Credentials
Before you can use a USB token, your system should meet the following requirements:

- A Cisco 871 router, Cisco 1800 series, Cisco 2800 series, a Cisco 3800 series router, or a Cisco 7200VXR NPE-G2 platform
- At least a Cisco IOS Release 12.3(14)T image running on any of the supported platforms
- A Cisco supported USB token (SafeNet/Aladdin eToken PRO 32 KB or 64 KB)
- A k9 image

Restrictions for Storing PKI Credentials

Restrictions for Specifying a Local Certificate Storage Location
When storing certificates to a local storage location, the following restrictions are applicable:

- Only local file systems may be used. An error message will be displayed if a remote file system is selected, and the command will not take effect.
- A subdirectory may be specified if supported by the local file system. NVRAM does not support subdirectories.

Restrictions for Specifying USB Token Storage
When using a USB token to store PKI data, the following restrictions are applicable:

- USB token support requires a 3DES (k9) Cisco IOS software image, which provides secure file storage.
- You cannot boot an image from a USB token. (However, you can boot a configuration from a USB token.)
- USB hubs are currently not supported. Thus, the number of supported devices is limited to the number of available USB ports.
Information About Storing PKI Credentials

Storing Certificates to a Local Storage Location

Certificates are stored to NVRAM by default; however, some routers do not have the required amount of NVRAM to successfully store certificates.

All Cisco platforms support NVRAM and flash local storage. Depending on your platform, you may have other supported local storage options including bootflash, slot, disk, USB flash, or USB token.

During run time, you can specify what active local storage device you would like to use to store certificates.

PKI Credentials and USB Tokens

To use a secure USB token on your router, you should understand the following concepts:

How a USB Token Works

A smart card is a small plastic card, containing a microprocessor and memory that allows you to store and process data. A USB token is a smart card with a USB interface. The token can securely store any type of file within its available storage space (32 KB). Configuration files that are stored on the USB token can be encrypted and accessed only via a user PIN. The device does not load the configuration file unless the proper PIN has been configured for secure deployment of device configuration files.

After you plug the USB token into the device, you must log into the USB token; thereafter, you can change default settings, such as the user PIN (default: 1234567890) and the allowed number of failed login attempts (default: 15 attempts) before future logins are refused. For more information on accessing and configuring the USB token, see the section "Logging Into and Setting Up the USB Token."

After you have successfully logged into the USB token, you can copy files from the device on to the USB token via the `copy` command. USB token RSA keys and associated IPsec tunnels remain available until the device is reloaded. To specify the length of time before the keys are removed and the IPsec tunnels are torn down, issue the `crypto pki token removal timeout` command. The default timeout is zero, which causes the RSA keys to be removed automatically after the eToken is removed from the device. The default appears in the running configuration as:

```
crypto pki token default removal timeout 0
```

The table below highlights the capabilities of the USB token.

### Table 12: Functionality Highlights for USB Tokens

<table>
<thead>
<tr>
<th>Function</th>
<th>USB Token</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessibility</td>
<td>Used to securely store and transfer digital certificates, preshared keys, and device configurations from the USB token to the device.</td>
</tr>
<tr>
<td>Storage Size</td>
<td>32 KB or 64 KB</td>
</tr>
<tr>
<td>Function</td>
<td>USB Token</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>File Types</td>
<td>• Typically used to store digital certificates, preshared keys, and device configurations for IPsec VPNs.</td>
</tr>
<tr>
<td></td>
<td>• USB tokens cannot store Cisco IOS images.</td>
</tr>
<tr>
<td>Security</td>
<td>• Files can be encrypted and accessed only with a user PIN.</td>
</tr>
<tr>
<td></td>
<td>• Files can also be stored in a nonsecure format.</td>
</tr>
<tr>
<td>Boot Configurations</td>
<td>• The device can use the configuration stored in the USB token during boot time.</td>
</tr>
<tr>
<td></td>
<td>• The device can use the secondary configuration stored in the USB token during boot time. (A secondary configuration allows users to load their IPsec configuration.)</td>
</tr>
</tbody>
</table>

**Benefits of USB Tokens**

USB token support on a Cisco router provides the following application benefits:

**Removable Credentials: Provide or Store VPN Credentials on an External Device for Deployment**

A USB token can use smart card technology to store a digital certificate and configuration for IPsec VPN deployment. This ability enhances the capability of the router to generate RSA public keys to authenticate at least one IPsec tunnel. (Because a router can initiate multiple IPsec tunnels, the USB token can contain several certificates, as appropriate.)

Storing VPN credentials on an external device reduces the threat of compromising secure data.

**PIN Configuration for Secure File Deployment**

A USB token can store a configuration file that can be used for enabling encryption on the router via a user-configured PIN. (That is, no digital certificates, preshared keys, or VPNs are used.)

**Touchless or Low Touch Configuration**

The USB token can provide remote software configuration and provisioning with little or no human interaction. Configuration is set up as an automated process. That is, the USB token can store a bootstrap configuration that the router can use to boot from after the USB token has been inserted into the router. The bootstrap configuration connects the router to a TFTP server, which contains a configuration that completely configures the router.
RSA Operations

A USB token may be used as a cryptographic device in addition to a storage device. Using a USB token as a cryptographic device allows RSA operations such as key generation, signing, and authentication to be performed on the token.

General-purpose, special-usage, encryption, or signature RSA key pairs with a modulus of 2048 bits or less may be generated from credentials located on your token storage device. Private keys are not distributed and remain on the token by default, however you may configure the private key storage location.

Keys that reside on a USB token are saved to persistent token storage when they are generated. Key deletion will remove the keys stored on the token from persistent storage immediately. (Keys that do not reside on a token are saved to or deleted from non-token storage locations when the write memory or a similar command is issued.)

Remote Device Configuration and Provisioning in a Secure Device Provisioning (SDP) Environment

SDP may be used to configure a USB token. The configured USB token may be transported to provision a device at a remote location. That is, a USB token may be used to transfer cryptographic information from one network device to another remote network device providing a solution for a staged USB token deployment.

For information about using USB tokens with SDP, see document titles in the “Additional References” section.

How to Configure PKI Storage

Specifying a Local Storage Location for Certificates

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki certificate storage location-name
4. exit
5. copy source-url destination-url
6. show crypto pki certificates storage

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enter your password if prompted.</td>
</tr>
</tbody>
</table>
### Purpose Command or Action

<table>
<thead>
<tr>
<th>Step 2</th>
<th>configure terminal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>crypto pki certificate storage  <em>location-name</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Device(config)# crypto pki certificate storage flash:/certs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Device(config)# exit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>copy  <em>source-url</em>  <em>destination-url</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Device# copy system:running-config nvram:startup-config</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>show crypto pki certificates storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Device# show crypto pki certificates storage</td>
</tr>
</tbody>
</table>

### Note
(Optional) Saves the running configuration to the startup configuration.

**Settings will only take effect when the running configuration is saved to the startup configuration.**

### Example

The following is sample output from the `show crypto pki certificates storage` command, which shows that the certificates are stored in the certs subdirectory of disk0:

```
Device# show crypto pki certificates storage
Certificates will be stored in disk0:/certs/
```
Setting Up and Using USB Tokens on Cisco Devices

Storing the Configuration on a USB Token

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `boot config usbtoken[0-9]:filename`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Specifies that the startup configuration file is stored in a secure USB token.</td>
</tr>
<tr>
<td><code>boot config usbtoken[0-9]:filename</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# boot config usbtoken0:file</td>
</tr>
</tbody>
</table>

Logging Into and Setting Up the USB Token

**How RSA Keys are Used with a USB Token**

- RSA keys are loaded after the USB token is successfully logged into the router.
- By default, newly generated RSA keys are stored on the most recently inserted USB token. Regenerated keys should be stored in the same location where the original RSA key was generated.

**Configuring the Device for Manual Login**

Unlike automatic login, manual login requires that the user know the actual USB token PIN.
Either the manual or automatic login is required.

Manual login can be used when storing a PIN on the device is not desirable. Manual login may also be suitable for some initial deployment or hardware replacement scenarios for which the device is obtained from the local supplier or drop-shipped to the remote site. Manual login can be executed with or without privileges, and it creates files and RSA keys on the USB token available to the Cisco IOS software. If a secondary configuration file is configured, it is executed only with the privileges of the user who is performing the login. Thus, if you want to use manual login and set up the secondary configuration on the USB token to perform anything useful, you need to enable privileges.

Manual login can also be used in recovery scenarios for which the device configuration has been lost. If the scenario contains a remote site that normally connects to the core network with a VPN, the loss of the configuration and RSA keys requires out-of-band services that the USB token can provide. The USB token can contain a boot configuration, a secondary configuration, or both, and RSA keys to authenticate the connection.

SUMMARY STEPS

1. enable
2. crypto pki token token-name [admin] login [pin]
3. show usbtoken 0-9:filename

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>enable</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>crypto pki token token-name [admin] login [pin]</td>
<td>Manually logs into the USB token. If the admin keyword is not specified initially you can re-enter the crypto pki token command again with this keyword option.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# crypto pki token usbtok0 admin login 5678</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>show usbtoken 0-9:filename</td>
<td>(Optional) Verifies whether the USB token has been logged on to the device.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# show usbtoken0:usbfile</td>
<td></td>
</tr>
</tbody>
</table>

What to Do Next

After you have logged into the USB token, it is available for use.
• To further configure the USB token, see the “Configuring the USB Token” section.

• To perform USB token administrative tasks, such as changing the user PIN, copying files from the router to the USB token set key storage location, and changing USB tokens, see the “Setting Administrative Functions on the USB Token” section.

Configuring the USB Token

After you have set up automatic login, you may perform this task to further configure the USB token.

PINs and Passphrases

For additional PIN security with automatic login, you may encrypt your PIN stored in NVRAM and set up a passphrase for your USB token. Establishing a passphrase allows you to keep your PIN secure; another user needs only to know the passphrase, not the PIN.

When the USB token is inserted into the device, the passphrase is needed to decrypt the PIN. Once the PIN is decrypted, the device can then use the PIN to log in to the USB token.

Note

The user needs a privilege level of 1 to log in.

Unlocking and Locking the USB Token

The USB token itself can be locked (encrypted) or unlocked (decrypted).

Unlocking the USB token allows it to be used. Once unlocked, Cisco IOS software treats the token as if it were automatically logged in. Any keys on the USB token are loaded, and if a secondary configuration file is on the token, it is executed with full user privileges (privilege level 15) independent of the privilege level of the logged-in user.

Locking the token, unlike logging out of the token, deletes any RSA keys loaded from the token and runs the secondary unconfiguration file, if configured.

Secondary Configuration and Unconfiguration Files

Configuration files that exist on a USB token are called secondary configuration files. If you create and configure a secondary configuration file, it is executed after the token is logged in. The existence of a secondary configuration file is determined by the presence of a secondary configuration file option in the Cisco IOS configuration stored in NVRAM. When the token is removed or logged out and the removal timer expires, a separate secondary unconfiguration file is processed to remove all secondary configuration elements from the running configuration. Secondary configuration and secondary unconfiguration files are executed at privilege level 15 and are not dependent on the level of the user logged in.
SUMMARY STEPS

1. enable
2. crypto pki token token-name unlock [pin]
3. configure terminal
4. crypto pki token token-name encrypted-user-pin [write]
5. crypto pki token token-name secondary unconfig file
6. exit
7. crypto pki token token-name lock [pin]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
</tbody>
</table>
| enable | Enables privileged EXEC mode.  
  • Enter your password if prompted. |
| **Example:** | |
| Device> enable | |
| **Step 2** | (Optional) Allows the token to be used if the USB token has been locked. |
| crypto pki token token-name unlock [pin] | Once unlocked, Cisco IOS software treats the token as if it has been automatically logged in. Any keys on the token are loaded and if a secondary configuration file exists, it is executed. |
| **Example:** | |
| Device# crypto pki token mytoken unlock mypin | |
| **Step 3** | Enters global configuration mode. |
| configure terminal | Enters global configuration mode. |
| **Example:** | |
| Device# configure terminal | |
| **Step 4** | (Optional) Encrypts the stored PIN in NVRAM. |
| crypto pki token token-name encrypted-user-pin [write] | (Optional) Encrypts the stored PIN in NVRAM. |
| **Example:** | |
| Device(config)# crypto pki token mytoken encrypted-user-pin write | |
| **Step 5** | (Optional) Specifies the secondary configuration file and its location. |
| crypto pki token token-name secondary unconfig file | (Optional) Specifies the secondary configuration file and its location. |
| **Example:** | |
| Device(config)# crypto pki token mytoken secondary unconfig configs/myunconfigfile.cfg | |
### Purpose

**Command or Action**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>exit</td>
<td>Enters privileged EXEC mode.</td>
</tr>
</tbody>
</table>

**Example:**  
Device(config)# exit

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>crypto pki token</td>
<td>(Optional) Deletes any RSA keys loaded from the token and runs the secondary unconfiguration file, if it exists.</td>
</tr>
</tbody>
</table>

**Example:**  
Device# crypto pki token mytoken lock mypin

### Examples

The following example shows both the configuration and encryption of a user PIN and then the device reloading and the user PIN being unlocked:

1. Configuring the user PIN
2. Encrypt the user PIN
3. Reloading the router.

#### Step 1: Configuring the user PIN

```
! Configuring the user PIN
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# crypto pki token usbtoken0: userpin
Enter password: mypassword
! Encrypt the user PIN
Device(config)# crypto pki token usbtoken0: encrypted-user-pin
Enter passphrase: mypassphrase
Device(config)# exit
Device#
```

#### Step 2: Encrypting the user PIN

```
Sep 20 21:51:38.076: %SYS-5-CONFIG_I: Configured from console by console
Device# show running config
crypto pki token usbtoken0 user-pin *encrypted*
! Reloading the router.
Device> enable
Password:
! Decrypting the user pin.
Device# crypto pki token usbtoken0: unlock
Token eToken is usbtoken0
Enter passphrase: mypassphrase
Token login to usbtoken0(eToken) successful
Device#
```

#### Step 3: Reloading the router

```
Sep 20 22:31:13.128: %CRYPTO-6-TOKENLOGIN: Cryptographic Token eToken
```

Login Successful

The following example shows how a secondary unconfiguration file might be used to remove secondary configuration elements from the running configuration. For example, a secondary configuration file might be used to set up a PKI trustpoint. A corresponding unconfiguration file, named mysecondaryunconfigfile.cfg, might contain this command line:

```plaintext
no crypto pki trustpoint token-tp
```

If the token were removed and the following commands executed, the trustpoint and associated certificates would be removed from the device's running configuration:

```plaintext
Device# configure terminal
Device(config)# no crypto pki token mytoken secondary unconfig mysecondaryunconfigfile.cfg
```

**What to Do Next**

After you have logged into and configured the USB token, it is available for use. If you want to perform USB token administrative tasks, such as changing the user PIN, copying files from the router to the USB token set key storage location, and changing USB tokens, see the “Setting Administrative Functions on the USB Token” section.

**Setting Administrative Functions on the USB Token**

Perform this task to change default settings, such as the user PIN, the maximum number of failed attempts on the USB token, or the credential storage location.

**SUMMARY STEPS**

1. enable
2. crypto pki token  *token-name*  admin | change-pin  *pin*
3. crypto pki token  *token-name*  device-name: label  *token-label*
4. configure terminal
5. crypto key storage  *device-name*:
6. crypto key generate rsa  *general-keys* | *usage-keys* | *signature* | *encryption* [ *label*  *key-label*] [ *exportable*] [ *modulus*  *modulus-size*] [ *storage*  *device-name*:] [ *redundancy*] [ *on*  *device-name*]:
7. crypto key move rsa  *key-label*  *non-exportable* | [ *on* | *storage*]  *location*
8. crypto pki token  *token-name* | *default* | removal timeout  *seconds*
9. crypto pki token  *token-name* | *default| max-retries  *number*
10. exit
11. copy usbflash[0-9]:*filename*  *destination-url*
12. show usbstoken[0-9]:*filename*
13. crypto pki token  *token-name*  logout

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
</tbody>
</table>
### Storing PKI Credentials

#### Setting Up and Using USB Tokens on Cisco Devices

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td>- Enter your password if prompted.</td>
</tr>
</tbody>
</table>

**Step 2**  
**crypto pki token**  
*token-name admin*  
*change-pin [pin]*  
(Optimal) Changes the user PIN number on the USB token.  
- If the PIN is not changed, the default PIN 1234567890 is used.  

**Note**  
After the PIN has been changed, you must reset the login failure count to zero (via the **crypto pki token max-retries** command). The maximum number of allowable login failures is set (by default) to 15.  

**Example:**  
Device# crypto pki token usbtoken0 admin change-pin

**Step 3**  
**crypto pki token**  
*token-name device-name:*  
*label token-label*  
(Optional) Sets or changes the name of the USB token.  
- The value of the *token-label* argument may be up to 31 alphanumeric characters in length including dashes and underscores.  

**Tip**  
This command is useful when configuring multiple USB tokens for automatic login, secondary configuration files, or other token specific settings.  

**Example:**  
Device# crypto pki token mytoken usb0: label newlabel

**Step 4**  
**configure terminal**  
Enter global configuration mode.  

**Example:**  
Device# configure terminal

**Step 5**  
**crypto key storage**  
*device-name:*  
(Optional) Sets the default RSA key storage location for newly created keys.  

**Note**  
Regardless of configuration settings, existing keys are stored on the device from where they were originally loaded.  

**Example:**  
Device(config)# crypto key storage usbtoken0:

**Step 6**  
**crypto key generate rsa**  
*general-keys | usage-keys | signature | encryption | [label key-label] | [exportable] [modulus modulus-size] [storage device-name:] [redundancy] [on device-name]:*  
(Optional) Generates the RSA key pair for the certificate server.  
- The *storage* keyword specifies the key storage location.  
- When specifying a label name by specifying the *key-label* argument, you must use the same name for the label that you plan to use for the certificate server (through the **crypto pki server cs-label** command). If a *key-label* argument is not specified, the default value, which is the fully qualified domain name (FQDN) of the device, is used.  

**Example:**  
Device(config)# crypto key generate rsa label tokenkey1 storage usbtoken0:

If the exportable RSA key pair is manually generated after the CA certificate has been generated, and before issuing the **no shutdown** command, then use the **crypto ca export pkcs12** command to export a PKCS12 file that contains the certificate server certificate and the private key.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>By default, the modulus size of a CA key is 1024 bits. The recommended modulus for a CA key is 2048 bits. The range for a modulus size of a CA key is from 350 to 4096 bits.</td>
<td></td>
</tr>
<tr>
<td>• The on keyword specifies that the RSA key pair is created on the specified device, including a Universal Serial Bus (USB) token, local disk, or NVRAM. The name of the device is followed by a colon (:).</td>
<td></td>
</tr>
<tr>
<td>Keys created on a USB token must be 2048 bits or less.</td>
<td></td>
</tr>
</tbody>
</table>

**Step 7**

**crypto key move rsa** *keylabel* [non-exportable | [on | storage] *location*  

**(Optional)** Moves existing Cisco IOS credentials from the current storage location to the specified storage location.

By default, the RSA key pair remains stored on the current device.

Generating the key on the device and moving it to the token takes less than a minute. Generating a key on the token, using the on keyword could take five to ten minutes, and is dependent on hardware key generation routines available on the USB token.

When an existing RSA key pair is generated in Cisco IOS, stored on a USB token, and used for an enrollment, it may be necessary to move those existing RSA key pairs to an alternate location for permanent storage.

This command is useful when using SDP with USB tokens to deploy credentials.

**Example:**

Device(config)# crypto key move rsa keypairname non-exportable on token

**Step 8**

**crypto pki token** {token-name | default} removal timeout [seconds]  

**(Optional)** Sets the time interval, in seconds, that the device waits before removing the RSA keys that are stored in the USB token after the USB token has been removed from the device.

**Note** If this command is not issued, all RSA keys and IPsec tunnels associated with the USB token are torn down immediately after the USB token is removed from the device.

**Example:**

Device(config)# crypto pki token usbtoken0 removal timeout 60

**Step 9**

**crypto pki token** {token-name | default} max-retries [number]  

**(Optional)** Sets the maximum number of consecutive failed login attempts allowed before access to the USB token is denied.

• By default, the value is set at 15.

**Example:**

Device(config)# crypto pki token usbtoken0 max-retries 20

**Step 10**

**exit**  

Exits global configuration mode.

**Example:**

Device(config)# exit

**Step 11**

**copy usbflash[0-9];filename destination-url**  

Copies files from USB token to the device.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: Device# copy usbflash0:file1 nvram:</td>
<td>• <em>destination-url</em>—See the <em>copy</em> command page documentation for a list of supported options.</td>
</tr>
<tr>
<td>Step 12 show usbtoken[0-9]:filename</td>
<td>(Optional) Displays information about the USB token. You can use this command to verify whether the USB token has been logged in to the device.</td>
</tr>
<tr>
<td>Example: Device# show usbtoken:usbfile</td>
<td></td>
</tr>
<tr>
<td>Step 13 crypto pki token token-name logout</td>
<td>Logs the device out of the USB token.</td>
</tr>
<tr>
<td>Example: Device# crypto pki token usbtoken0 logout</td>
<td>Note If you want to save any data to the USB token, you must log back into the token.</td>
</tr>
</tbody>
</table>

### Troubleshooting USB Tokens

This section contains descriptions of the following Cisco IOS commands that can be used to help troubleshoot possible problems that may arise while using a USB token:

#### Troubleshooting the USB Port Connection

Use the *show file systems* command to determine whether the router recognizes that there is a USB module plugged into a USB port. The USB module should appear on the list of file systems. If the module does not appear on the list, it can indicate any of the following problems:

- A connection problem with the USB module.
- The Cisco IOS image running on the router does not support a USB module.
- A hardware problem with the USB module itself.

Sample output from the *show file systems* command showing a USB token appears below. The USB module listing appears in the last line of the examples.

```
Device# show file systems
File Systems:
<table>
<thead>
<tr>
<th>Size(b)</th>
<th>Free(b)</th>
<th>Type</th>
<th>Flags</th>
<th>Prefixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>opaque</td>
<td>rw</td>
<td>archive:</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>opaque</td>
<td>rw</td>
<td>system:</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>opaque</td>
<td>rw</td>
<td>null:</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>network</td>
<td>rw</td>
<td>tftp:</td>
</tr>
</tbody>
</table>
* 129880064 69414912 disk rw flash:| |
491512 486395 nvr | nvram rw nvr: |
- - opaque wo syslog: |
- - opaque rw xmodem: |
- - opaque rw ymodem: |
- - network rw rcp: |
```
Determining if a USB Token is Supported by Cisco

Use the `show usb device` command to determine if a USB token is supported by Cisco. The following output from this command indicates whether or not the module is supported is bold in the sample output below:

```
Router# show usb device
Host Controller:1
Address:0x11
Device Configured:YES
Device Supported:YES
Description:eToken Pro 4254
Manufacturer:AKS
Version:1.0
Serial Number:
Device Handle:0x1010000
USB Version Compliance:1.0
Class Code:0xFF
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x529
Product ID:0x514
Max. Packet Size of Endpoint Zero:8
Number of Configurations:1
Speed:Low
Selected Configuration:1
Selected Interface:0
Configuration:
  Number:1
  Number of Interfaces:1
  Description:
  Attributes:None
  Max Power:60 mA
  Interface:
    Number:0
    Description:
    Class Code:255
    Subclass:0
    Protocol:0
    Number of Endpoints:0
```

Determining USB Token Device Problems

Use the `show usb controllers` command to determine if there is a hardware problem with a USB flash module. If the `show usb controllers` command displays an error, the error indicates a hardware problem in the USB module.

You can also use the `show usb controllers` command to verify that copy operations onto a USB flash module are occurring successfully. Issuing the `show usb controllers` command after performing a file copy should display successful data transfers.

The following sample output for the `show usb controllers` command displays a working USB flash module:

```
Router# show usb controllers
Name:1362HCD
Controller ID:1
```
Controller Specific Information:
Revision:0x11
Control:0x80
Command Status:0x0
Hardware Interrupt Status:0x24
Hardware Interrupt Enable:0x80000040
Hardware Interrupt Disable:0x80000040
Frame Interval:0x27782EDF
Frame Remaining:0x13C1
Frame Number:0xDA4C
LSThreshold:0x628
RhDescriptorA:0x19000202
RhDescriptorB:0x0
RhStatus:0x0
RhPort1Status:0x100103
RhPort2Status:0x100303
Hardware Configuration:0x3029
DMA Configuration:0x0
Transfer Counter:0x1
Interrupt:0x9
Interrupt Enable:0x196
Chip ID:0x3630
Buffer Status:0x0
Direct Address Length:0x80A00
ATL Buffer Size:0x600
ATL Buffer Port:0x0
ATL Block Size:0x100
ATL PTD Skip Map:0xFFFFFFFF
ATL PTD Last:0x20
ATL Current Active PTD:0x0
ATL Threshold Count:0x1
ATL Threshold Timeout:0xFF
Int Level:1
Transfer Completion Codes:
Success :920 CRC :0
Bit Stuff :0 Stall :0
No Response :0 Overrun :0
Underrun :0 Other :0
Buffer Overrun :0 Buffer Underrun :0
Transfer Errors:
Canceled Transfers :2 Control Timeout :0
Transfer Failures:
Interrupt Transfer :0 Bulk Transfer :0
Isochronous Transfer :0 Control Transfer:0
Transfer Successes:
Interrupt Transfer :0 Bulk Transfer :26
Isochronous Transfer :0 Control Transfer:894
USBD Failures:
Enumeration Failures :0 No Class Driver Found:0
Power Budget Exceeded:0
USB MSCD SCSI Class Driver Counters:
Good Status Failures :3 Command Fail :0
Good Status Timed out:0 Device not Found:0
Device Never Opened :0 Drive Init Fail :0
Illegal App Handle :0 Bad API Command :0
Invalid Unit Number :0 Invalid Argument:0
Application Overflow :0 Device in use :0
Control Pipe Stall :0 Malloc Error :0
Device Stalled :0 Bad Command Code:0
Device Detached :0 Unknown Error :0
Chip Invalid Logic Unit Num:0
USB Aladdin Token Driver Counters:
Token Inserted :1 Token Removed :0
Send Insert Msg Fail :0 Response Txns :434
Dev Entry Add Fail :0 Request Txns :434
Dev Entry Remove Fail:0 Response Txn Fail :0
Response Txn Fail :0 Command Txn Fail:0
Txn Invalid Dev Handle:0
USB Flash File System Counters:
Flash Disconnected :0 Flash Connected :1
Flash Device Fail :0 Flash OK :1
Flash startstop Fail :0 Flash FS Fail :0
USB Secure Token File System Counters:
Displaying USB Token Information

Use the `dir` command with the `filesystem` keyword option `usbtoken0-9` to display all files, directories, and their permission strings on the USB token.

The following sample output displays directory information for the USB token:

```
Device# dir usbtoken1:
Directory of usbtoken1:/
  2 d--- 64 Dec 22 2032 05:23:40 +00:00 1000
  5 d--- 4096 Dec 22 2032 05:23:40 +00:00 1001
  8 d--- 0 Dec 22 2032 05:23:40 +00:00 1002
 10 d--- 512 Dec 22 2032 05:23:42 +00:00 1003
 12 d--- 0 Dec 22 2032 05:23:42 +00:00 5000
 13 d--- 0 Dec 22 2032 05:23:42 +00:00 6000
 14 d--- 0 Dec 22 2032 05:23:42 +00:00 7000
 15 ---- 940 Jun 27 1992 12:50:42 +00:00 mystartup-config
 16 ---- 1423 Jun 27 1992 12:51:14 +00:00 myrunning-config
32768 bytes total (858 bytes free)
```

The following sample output displays directory information for all devices to which the device is aware:

```
Device# dir allfilesystems
Directory of archive:/
No files in directory
No space information available
Directory of system:/
  2 drwx 0 <no date> its
 115 dr-x 0 <no date> lib
 144 dr-x 0 <no date> memory
  1 -rw- 1906 <no date> running-config
 114 dr-x 0 <no date> vfiles
No space information available
Directory of flash:/
  1 -rw- 30125020 Dec 22 2032 03:06:04 +00:00 c3825-entservicesk9-mz.123-14.T
129880064 bytes total (99753984 bytes free)
Directory of nvram:
 476 -rw- 1947 <no date> startup-config
 477 ---- 46 <no date> private-config
 478 -rw- 1947 <no date> underlying-config
  1 -rw- 0 <no date> ifIndex-table
  2 ---- 4 <no date> rf_cold_starts
  3 ---- 14 <no date> persistent-data
491512 bytes total (486395 bytes free)
Directory of usbspand:
  1 -rw- 30125020 Dec 22 2032 05:31:32 +00:00 c3825-entservicesk9-mz.123-14.T
63158272 bytes total (33033216 bytes free)
Directory of usbtoken1:
  2 d--- 64 Dec 22 2032 05:23:40 +00:00 1000
  5 d--- 4096 Dec 22 2032 05:23:40 +00:00 1001
  8 d--- 0 Dec 22 2032 05:23:40 +00:00 1002
 10 d--- 512 Dec 22 2032 05:23:42 +00:00 1003
 12 d--- 0 Dec 22 2032 05:23:42 +00:00 5000
 13 d--- 0 Dec 22 2032 05:23:42 +00:00 6000
 14 d--- 0 Dec 22 2032 05:23:42 +00:00 7000
 15 ---- 940 Jun 27 1992 12:50:42 +00:00 mystartup-config
 16 ---- 1423 Jun 27 1992 12:51:14 +00:00 myrunning-config
32768 bytes total (858 bytes free)
```
Configuration Examples for PKI Storage

Example: Storing Certificates to a Specific Local Storage Location

The following configuration example shows how to store certificates to the certs subdirectory. The certs subdirectory does not exist and is automatically created.

```plaintext
Router# dir nvram:
114 -rw- 4687 <no date> startup-config
115 ---- 5545 <no date> private-config
116 -rw- 4687 <no date> underlying-config
 1 ---- 34 <no date> persistent-data
 3 -rw- 707 <no date> ioscaroot#7401CA.cer
 9 -rw- 863 <no date> msca-root#826E.cer
10 -rw- 759 <no date> msca-root#1BA8CA.cer
11 -rw- 863 <no date> msca-root#75B8.cer
24 -rw- 1149 <no date> storagename#6500CA.cer
26 -rw- 863 <no date> msca-root#83EE.cer
129016 bytes total (92108 bytes free)

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# crypto pki certificate storage disk0:/certs
Requested directory does not exist -- created
Certificates will be stored in disk0:/certs/
Router(config)# end
Router# write
*May 27 02:09:00:%SYS-5-CONFIG_I:Configured from console by consolemem
Building configuration...
[OK]
Router# directory disk0:/certs
Directory of disk0:/certs/
 14 -rw- 707 May 27 2005 02:09:02 00:00 ioscaroot#7401CA.cer
15 -rw- 863 May 27 2005 02:09:02 00:00 msca-root#826E.cer
16 -rw- 759 May 27 2005 02:09:02 00:00 msca-root#1BA8CA.cer
17 -rw- 863 May 27 2005 02:09:02 00:00 msca-root#75B8.cer
18 -rw- 1149 May 27 2005 02:09:02 00:00 storagename#6500CA.cer
19 -rw- 863 May 27 2005 02:09:02 00:00 msca-root#83EE.cer
47894528 bytes total (20934656 bytes free)
! The certificate files are now on disk0/certs:
```

Example: Logging Into a USB Token and Saving RSA Keys to the USB Token

The following configuration example shows how to log in to the USB token, generate RSA keys, and store the RSA keys on the USB token:

```plaintext
! Configure the router to automatically log into the eToken
configure terminal
  crypto pki token default user-pin 0 1234567890
! Generate RSA keys and enroll certificates with the CA.
  crypto pki trustpoint IOSCA
  enrollment url http://10.23.2.2
  exit
  crypto ca authenticate IOSCA
Certificate has the following attributes:
  Fingerprint MD5:23272BD4 37E3D9A4 236F7E1A F534444E
  Fingerprint SHA1:D1B4D9F8 D603249A 793B3CAF 8342E1FE 3934EB7A
% Do you accept this certificate? [yes/no]:yes
Trustpoint CA certificate accepted.
  crypto pki enroll
  crypto pki enroll IOSCA
%```
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this password to the CA Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration.
Please make a note of it.
Password:
Re-enter password:
% The subject name in the certificate will include: c2851-27.cisco.com
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The 'show crypto ca certificate IOSCA verbose' command will show the fingerprint.
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint MD5:E6DDAB1B 0E30EEF6 54529DBA DA787DBA
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint SHA1:3B0F33B 7B7C02A10 3935042B 04B6CD3D 61039251
*Jan 13 06:47:21.021:%PKI-6-CERTRET:Certificate received from Certificate Authority
! Issue the write memory command, which will automatically save the RSA keys to the eToken
! instead of private NVRAM.
Router# write memory
Building configuration...
[OK]
*Jan 13 06:47:29.8481:%CRYPTO-6-TOKENSTOREKEY:Key c2851-27.cisco.com stored on Cryptographic Token eToken Successfully
The following sample output from the show crypto key mypubkey rsa command displays stored credentials after they are successfully loaded from the USB token. Credentials that are stored on the USB token are in the protected area. When storing the credentials on the USB token, the files are stored in a directory called /keystore. However, the key files are hidden from the command-line interface (CLI).

Router#
show crypto key mypubkey rsa
% Key pair was generated at:06:37:26 UTC Jan 13 2005
Key name:c2851-27.cisco.com
Usage:General Purpose Key
Key is not exportable.
Key Data:
732EDF4E 3CA0CDA8 387ABF05 E88B22F2 21431F1AE 5051FEE3 FCDFA934 3FBD3603
7C977B54 B5E999BF 7FC30217 F445AF68 A4BA2ED6 172D3D09 B5020301 0001
% Key pair was generated at:06:37:27 UTC Jan 13 2005
Key name:c2851-27.cisco.com.server
Usage:Encryption Key
Key is not exportable.
Key Data:
305C3000 06092A86 4886F70D 01010105 00034B00 30480241 00E3C644 43AA7DDD
732EDF4E 3CA0CDA8 387ABF05 E88B22F2 21431F1AE 5051FEE3 FCDFA934 3FBD3603
7C977B54 B5E999BF 7FC30217 F445AF68 A4BA2ED6 172D3D09 B5020301 0001

Additional References

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connecting the USB modules to the router</td>
<td>Cisco Access Router USB Flash Module and USB eToken Hardware Installation Guide</td>
</tr>
<tr>
<td>eToken and USB flash data sheet</td>
<td>USB eToken and USB Flash Features Support</td>
</tr>
<tr>
<td>RSA keys</td>
<td>Deploying RSA Keys Within a PKI</td>
</tr>
</tbody>
</table>
Feature Information for Storing PKI Credentials

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.
### Table 13: Feature Information for Storing PKI Credentials

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| USB Token and Secure Device Provisioning (SDP) Integration | 12.4(15)T | This feature provides the ability to provision remote devices with USB tokens using SDP. The following sections in this document provide information about this feature:  
- Benefits of USB Tokens  
- Setting Administrative Functions on the USB Token  
The following commands were introduced by this feature: `binary file`, `crypto key move rsa`, `template file`.  
**Note** This document introduces the benefits of using USB tokens and SDP for a deployment solution. |
| Cisco IOS USB Token PKI Enhancements -- Phase 2 | 12.4(11)T | This feature enhances USB token functionality by using the USB token as a cryptographic device. USB tokens may be used for RSA operations such as key generation, signing, and authentication. The following sections in this document provide information about this feature:  
- Benefits of USB Tokens  
- Logging Into and Setting Up the USB Token  
- Setting Administrative Functions on the USB Token  
**Note** This document introduces the benefits of using USB tokens and the keys on the token for RSA operations. |
USB Storage PKI Enhancements

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12.4(4)T</td>
<td>This feature enhances the USB token PIN security for automatic login and increases the flexibility of USB token configuration and the RSA key storage. Cisco IOS Release 12.4(11)T introduced support for USB Storage on NPE-G2. The following sections provide information about this feature:</td>
</tr>
<tr>
<td></td>
<td>12.4(11)T</td>
<td>• Configuring the USB Token</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Setting Administrative Functions on the USB Token</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following commands were introduced or modified by this feature: crypto key storage, crypto pki generate rsa, crypto pki token encrypted-user-pin, crypto pki token label, crypto pki token lock, crypto pki token secondary unconfig, crypto pki token unlock</td>
</tr>
</tbody>
</table>
## Feature Information for Storing PKI Credentials

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Certificate -- Storage Location Specification | 12.2(33)SXH, 12.2(33)SRA, 12.4(2)T | This feature allows you to specify the storage location of local certificates for platforms that support storing certificates as separate files. All Cisco platforms support NVRAM, which is the default location, and flash local storage. Depending on your platform, you may have other supported local storage options including bootflash, slot, disk, USB flash, or USB token. The following sections provide information about this feature:

- Storing Certificates to a Local Storage Location
- Specifying a Local Storage Location for Certificates
- Storing Certificates to a Specific Local Storage Location Example

The following commands were introduced by this feature: `crypto pki certificate storage`, `show crypto pki certificates storage`
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| USB Storage                                      | 12.3(14)T   | This feature enables certain models of Cisco routers to support USB tokens. USB tokens provide secure configuration distribution and allow users to VPN credentials for deployment. Cisco IOS Release 12.4(11)T introduced support for USB Storage on NPE-G2. The following sections provide information about this feature:  
  • PKI Credentials and USB Tokens  
  • Setting Up and Using USB Tokens on Cisco Routers  
  • Troubleshooting USB Tokens  
  • Logging Into a USB Token and Saving RSA Keys to the USB Token Example  
  The following commands were introduced or modified by this feature: copy, crypto pki token change-pim, crypto pki token login, crypto pki token logout, crypto pki token max-retries, crypto pki token removal timeout, crypto pki token secondary config, crypto pki token user-pin, debug usb driver, dir, show usb controllers, show usb device, show usb driver, show usbtoken |
|                                                 | 12.4(11)T   |                                                                                                                                                   |
| RSA 4096-bit Key Generation in Software Crypto Engine Support | 15.1(1)T    | The range value for the modulus keyword value for the crypto key generate rsa command is extended from 360 to 2048 bits to 360 to 4096 bits. |

**Feature Information for Storing PKI Credentials**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| USB Storage                                      | 12.3(14)T   | This feature enables certain models of Cisco routers to support USB tokens. USB tokens provide secure configuration distribution and allow users to VPN credentials for deployment. Cisco IOS Release 12.4(11)T introduced support for USB Storage on NPE-G2. The following sections provide information about this feature:  
  • PKI Credentials and USB Tokens  
  • Setting Up and Using USB Tokens on Cisco Routers  
  • Troubleshooting USB Tokens  
  • Logging Into a USB Token and Saving RSA Keys to the USB Token Example  
  The following commands were introduced or modified by this feature: copy, crypto pki token change-pim, crypto pki token login, crypto pki token logout, crypto pki token max-retries, crypto pki token removal timeout, crypto pki token secondary config, crypto pki token user-pin, debug usb driver, dir, show usb controllers, show usb device, show usb driver, show usbtoken |
|                                                 | 12.4(11)T   |                                                                                                                                                   |
| RSA 4096-bit Key Generation in Software Crypto Engine Support | 15.1(1)T    | The range value for the modulus keyword value for the crypto key generate rsa command is extended from 360 to 2048 bits to 360 to 4096 bits. |
Source Interface Selection for Outgoing Traffic with Certificate Authority

The Source Interface Selection for Outgoing Traffic with Certificate Authority feature allows the IP address of an interface to be specified and used as the source address for all outgoing TCP connections associated with that trustpoint when a designated trustpoint has been configured.

- Finding Feature Information, page 285
- Information About Source Interface Selection for Outgoing Traffic with Certificate Authority, page 286
- How to Configure Source Interface Selection for Outgoing Traffic with Certificate Authority, page 286
- Configuration Examples for Source Interface Selection for Outgoing Traffic with Certificate Authority, page 289
- Additional References, page 290
- Feature Information for Source Interface Selection for Outgoing Traffic with Certificate Authority, page 291
- Glossary, page 292

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.
Information About Source Interface Selection for Outgoing Traffic with Certificate Authority

Certificates That Identify an Entity

Certificates can be used to identify an entity. A trusted server, known as the certification authority (CA), issues the certificate to the entity after determining the identity of the entity. A router that is running Cisco IOS software obtains its certificate by making a network connection to the CA. Using the Simple Certificate Enrollment Protocol (SCEP), the router transmits its certificate request to the CA and receives the granted certificate. The router obtains the certificate of the CA in the same manner using SCEP. When validating a certificate from a remote device, the router may again contact the CA or a Lightweight Directory Access Protocol (LDAP) or HTTP server to determine whether the certificate of the remote device has been revoked. (This process is known as checking the certificate revocation list [CRL].)

In some configurations, the router may make the outgoing TCP connection using an interface that does not have a valid or IP address that can be routed. The user must specify that the address of a different interface be used as the source IP address for the outgoing connection. Cable modems are a specific example of this requirement because the outgoing cable interface (the RF interface) usually does not have an IP address that can be routed. However, the user interface (usually Ethernet) does have a valid IP address.

Source Interface for Outgoing TCP Connections Associated with a Trustpoint

The `crypto ca trustpoint` command is used to specify a trustpoint. The `source interface` command is used along with the `crypto ca trustpoint` command to specify the address of the interface that is to be used as the source address for all outgoing TCP connections associated with that trustpoint.

Note

If the interface address is not specified using the `source interface` command, the address of the outgoing interface is used.

How to Configure Source Interface Selection for Outgoing Traffic with Certificate Authority

Configuring the Interface for All Outgoing TCP Connections Associated with a Trustpoint

Perform this task to configure the interface that you want to use as the source address for all outgoing TCP connections associated with a trustpoint.
SUMMARY STEPS

1. enable
2. configure terminal
3. crypto ca trustpoint name
4. enrollment [mode] [retry period minutes] [retry count number] url url [pem]
5. source interface interface-address
6. interface type slot / port
7. description string
8. ip address ip-address mask
9. interface type slot/port
10. description string
11. ip address ip-address mask
12. crypto map map-name

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>crypto ca trustpoint name</td>
<td>Declares the Certificate Authority (CA) that your router should use and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router (config)# crypto ca trustpoint ms-ca</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>enrollment [mode] [retry period minutes] [retry count number] url url [pem]</td>
<td>Specifies the following enrollment parameters of the CA:</td>
</tr>
</tbody>
</table>
| Example: | Router (ca-trustpoint)# enrollment url http://caserver.myexample.com or- Router (ca-trustpoint)# enrollment url http://[2001:DB8:1:1::1]:80 | • (Optional) The mode keyword specifies the registration authority (RA) mode, if your CA system provides an RA. By default, RA mode is disabled.  
• (Optional) The retry period keyword and minutes argument specifies the period, in minutes, in which the router waits before sending the CA another certificate request. Valid values are from 1 to 60. The default is 1. |
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong> source interface interface-address</td>
<td>Interface to be used as the source address for all outgoing TCP connections associated with that trustpoint.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router (ca-trustpoint)# interface ethernet 0</td>
</tr>
<tr>
<td><strong>Step 6</strong> interface type slot / port</td>
<td>Configures an interface type and enters interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router (ca-trustpoint)# interface ethernet 1</td>
</tr>
<tr>
<td><strong>Step 7</strong> description string</td>
<td>Adds a description to an interface configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router (config-if)# description inside interface</td>
</tr>
<tr>
<td><strong>Step 8</strong> ip address ip-address mask</td>
<td>Sets a primary or secondary IP address for an interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router (config-if)# ip address 10.1.1.1 255.255.255.0</td>
</tr>
<tr>
<td><strong>Step 9</strong> interface type slot/port</td>
<td>Configures an interface type.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router (config-if)# interface ethernet1/0</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Adds a description to an interface configuration.</td>
</tr>
<tr>
<td><strong>description string</strong></td>
<td>Example: Router (config-if)# description outside interface 10.1.1.205 255.255.255.0</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>Sets a primary or secondary IP address for an interface.</td>
</tr>
<tr>
<td><strong>ip address ip-address mask</strong></td>
<td>Example: Router (config-if)# ip address 10.2.2.205 255.255.255.0</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>Applies a previously defined crypto map set to an interface.</td>
</tr>
<tr>
<td><strong>crypto map map-name</strong></td>
<td>Example: Router (config-if)# crypto map mymap</td>
</tr>
</tbody>
</table>

**Troubleshooting Tips**

Ensure that the interface specified in the command has a valid address. Attempt to ping the router using the address of the specified interface from another device (possibly the HTTP or LDAP server that is serving the CRL). You can do the same thing by using a traceroute to the router from the external device.

You can also test connectivity between the router and the CA or LDAP server by using Cisco IOS command-line interface (CLI). Enter the `ping ip` command and respond to the prompts. If you answer "yes" to the "Extended commands [n]:" prompt, you can specify the source address or interface.

In addition, you can use Cisco IOS CLI to input a `traceroute ip` command. If you enter the `traceroute ip` command (in EXEC mode), you are prompted for the destination and source address. You should specify the CA or LDAP server as the destination and the address of the interface that you specified in the "source interface" as the source address.

**Configuration Examples for Source Interface Selection for Outgoing Traffic with Certificate Authority**

**Source Interface Selection for Outgoing Traffic with Certificate Authority Example**

In the following example, the router is located in a branch office. The router uses IP Security (IPSec) to communicate with the main office. Ethernet 1 is the "outside" interface that connects to the Internet Service...
Provider (ISP). Ethernet 0 is the interface connected to the LAN of the branch office. To access the CA server located in the main office, the router must send its IP datagrams out interface Ethernet 1 (address 10.2.2.205) using the IPSec tunnel. Address 10.2.2.205 is assigned by the ISP. Address 10.2.2.205 is not a part of the branch office or main office.

The CA cannot access any address outside the company because of a firewall. The CA sees a message coming from 10.2.2.205 and cannot respond (that is, the CA does not know that the router is located in a branch office at address 10.1.1.1, which it is able to reach).

Adding the source interface command tells the router to use address 10.1.1.1 as the source address of the IP datagram that it sends to the CA. The CA is able to respond to 10.1.1.1.

This scenario is configured using the source interface command and the interface addresses as described above.

crypto ca trustpoint ms-ca
   enrollment url http://ms-ca:80/certsrv/mscep/mscep.dll
   source interface ethernet0
interface ethernet 0
description inside interface
ip address 10.1.1.1 255.255.255.0
interface ethernet 1
description outside interface
ip address 10.2.2.205 255.255.255.0
crypto map main-office

Additional References

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
<tr>
<td>Configuring IPSec and certification authority</td>
<td>Security for VPNs with IPsec</td>
</tr>
<tr>
<td>IPSec and certification authority commands</td>
<td>Cisco IOS Security Command Reference</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None.</td>
<td>To locate and download MIBs for selected platforms,</td>
</tr>
<tr>
<td></td>
<td>Cisco software releases, and feature sets, use Cisco</td>
</tr>
<tr>
<td></td>
<td>MIB Locator found at the following URL:</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

Feature Information for Source Interface Selection for Outgoing Traffic with Certificate Authority

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 14: Feature Information for Source Interface Selection for Outgoing Traffic with Certificate Authority

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Interface Selection for Outgoing Traffic with Certificate Authority</td>
<td>12.2(15)T</td>
<td>This feature allows the IP address of an interface to be specified and used as the source address for all outgoing TCP connections associated with that trustpoint when a designated trustpoint has been configured. This feature was introduced in Cisco IOS Release 12.2(15)T. The following command was introduced or modified: <strong>source interface</strong>.</td>
</tr>
<tr>
<td>PKI IPv6 Support for VPN Solutions</td>
<td>15.2(1)T</td>
<td>The <strong>enrollment url (ca-trustpoint)</strong> command was modified to specify an IPv6 address in the CA URL.</td>
</tr>
</tbody>
</table>
authenticate—to prove the identity of an entity using the certificate of an identity and a secret that the identity poses (usually the private key corresponding to the public key in the certificate).

CA—Certificate Authority. A CA is an entity that issues digital certificates (especially X.509 certificates) and vouches for the binding between the data items in a certificate.

CA authentication—The user manually approves a certificate from a root CA. Usually a fingerprint of the certificate is presented to the user, and the user is asked to accept the certificate based on the fingerprint. The certificate of a root CA is signed by itself (self-signed) so that it cannot be automatically authenticated using the normal certificate verification process.

CRL—certificate revocation list. A CRL is a data structure that enumerates digital certificates that have been invalidated by their issuer prior to when they were scheduled to expire.

enrollment—A router receives its certificate through the enrollment process. The router generates a request for a certificate in a specific format (known as PKCS #10). The request is transmitted to a CA, which grants the request and generates a certificate encoded in the same format as the request. The router receives the granted certificate and stores it in an internal database for use during normal operations.

certificate—A data structure defined in International Organization for Standardization (ISO) standard X.509 to associate an entity (machine or human) with the public key of that entity. The certificate contains specific fields, including the name of the entity. The certificate is normally issued by a CA on behalf of the entity. In this case the router acts as its own CA. Common fields within a certificate include the distinguished name (DN) of the entity, the DN of the authority issuing the certificate, and the public key of the entity.

LDAP—Lightweight Directory Access Protocol. A LDAP is a protocol that provides access for management and browser applications that provide read-and-write interactive access to the X.500 directory.
INDEX

H
HTTP - source interface selection 286
  source interface for outgoing TCP connections 286

P
PKI integration with AAA server 46
  configuring 46

S
source interface selection for outgoing traffic with Certificate Authority 286, 289
  certificates that identify an entity 286
  configuring 286
  example 289
  troubleshooting 289